THE USENIX ASSOCIATION NEWSLETTER

;login:

OCTOBER 1996 • VOLUME 21 • NUMBER 5

CONTENTS

EDITORIAL
Journalism 3

LETTERS TO THE EDITOR 4

USENIX NEWS
Letter from the President *by Andrew Hume* 7
Reports on the Sixth USENIX Security Symposium 8
Reports on the Fourth Annual Tcl/Tk Workshop 15
The Demise of *Computing Systems* *by Andrew Hume* 17
In Search of a Better Internet Language *by John E. Schimmel* 17
IPv6 & IPsec Software Available *by Ran Atkinson* 18
CMU Andrew Consortium Releases C++ Version of Andrew UI and WP System *by Anne Watzman* 18
International Olympiad of Informatics Results . 19

SAGE NEWS
Not Fancy, Just Functional *by Tina M. Darmohray* 20
Jack of All Trades *by Eric Berglund* 21
Watching Browser Technologies *by Shawn Instenes* 22
Perl Practicum: Network Wiles (Part II) *by Hal Pomeranz* 23
New SAGE Booklet Now Available 26
Bay Lisa Meeting News 26

FEATURES
Defamation and the Net
A new column on legal issues *by Nicholas S. Nassif* 27
Are Standards Worth the Effort? *by Nick Stoughton* 29
Is Your Domain Name System Secure? *by John Gilmore* 31
The Webmaster: Web Site Memories with Cookies *by Dave Taylor* 32
The Importance of Documentation *by Scott Hazen Mueller* 35
Language and Communication *by Peter Salus* ..37
Using Java
A new series of articles about Java by Rik Farrow 39
Webstar Award Competition 42

STANDARDS REPORTS
An Update on Standards Relevant to USENIX Members *by Nicholas M. Stoughton* 43

BOOK REVIEWS
The Bookworm *by Peter H. Salus* 45
The Java Programming Language –Reviewed by George W. Leach 46
Exploring Java–Reviewed by Rik Farrow 46
HTML: The Definitive Guide –Reviewed by George W. Leach 47
Cultural Treasures of the Internet –Reviewed by George W. Leach 48

ANNOUNCEMENTS & CALLS
OSDI '96 50
2nd Workshop on Electronic Commerce 52
1997 USENIX Technical Conference 54
7th Conference on Computers, Freedom and Privacy 57
COOTS '97 58
LISA '97 60
7th USENIX Security Symposium 62
15th IEEE Symposium on Reliable Distributed Systems 64

ETC . . .
Local User Groups 72
Calendar of Events 74

USENIX The UNIX and Advanced Computing Systems Professional and Technical Association

;login: is the official newsletter of the USENIX Association.

- *;login:* (ISSN 1044-6397) Volume 21, Number 5 (October 1996) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.
- $25 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for nonmembers are $50 per year.
- Periodicals postage paid at Berkeley, CA and additional offices.
- POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

EDITORIAL STAFF

- Rob Kolstad, Editor <*kolstad@usenix.org*>
- Tina Darmohray, SAGE News Editor <*tmd@usenix.org*>
- Nick Stoughton, Standards Report Editor <*nick@usenix.org*>
- Pennfield Jensen, Managing Editor <*penn@usenix.org*>
- Sylvia Stein Wright, Copy Editor
- Rhea Gossett, Proofreader

ADVERTISING

- Zanna Knight <*zanna@usenix.org*>

MEMBERSHIP AND PUBLICATIONS

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Phone: 510 528 8649
Fax: 510 548 5738
Email: <*office@usenix.org*>
WWW: <*http://www.usenix.org*>

CONTRIBUTIONS SOLICITED

You are encouraged to contribute articles, book reviews, and announcements to *;login:*. Send them via email to <*login@usenix.org*> or through the postal system to the Association office. Send SAGE material to <*tmd@usenix.org*>. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

© 1996 USENIX Association. USENIX is a registered trademark of the USENIX Association. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited. The Association acknowledges all other trade references made herein.

;login: is produced with Framemaker 5.0 software provided by Frame Technology. The system is served by a Sun SPARCsystem 10 server. Final output is generated by a 600 dpi QMS 860 laser printer, donated by Quality Micro Systems. *;login:* is printed on recycled paper.

The closing dates for submissions to the next two issues of *;login:* are October 14 and December 10, 1996.

SAVE THESE DATES!

Upcoming USENIX Events

2nd Symposium on Operating Systems Design and Implementation (OSDI '96)
October 28–31, 1996, Seattle, WA
Co-sponsored by ACM SIGOPS and IEEE TCOS
Program Chairs: Karin Petersen, *Xerox PARC,* Willy Zwaenepoel, *Rice University*

2nd Workshop on Electronic Commerce
November 18-20 1996, Claremont Hotel, Oakland, CA
Co-sponsored by the Fisher Center for Information Technology Management, UC Berkeley, and the School of Information Management and Systems, UC Berkeley
Program Chair: Doug Tygar, *Carnegie Mellon University*

USENIX 1997 Annual Technical Conference
January 6-10, 1997, Anaheim Marriott, Anaheim, CA
Program Chair: John Kohl, *Atria Software*
Invited Talks Coordinators: Mary Baker, *Stanford,* Berry Kercheval, *Xerox PARC*
Camera-ready Papers Due: November 13, 1996

USELINUX: Linux Applications Development and Deployment Conference
January 6-10, 1997, Anaheim Marriott, Anaheim, CA
Co-located with the USENIX Annual Technical Conference
Conference Chairs: Michael K. Johnson, *Red Hat Software,* John "maddog" Hall, *DEC*

6th System Administration, Networking & Security Conference (SANS)
April 21-26, 1997, Baltimore Inner Harbor, MD. Sponsored by Open Systems Conference Board with the assistance of SAGE, The Systems Administrators Guild

3rd USENIX Conference on Object-Oriented Technologies and Systems (COOTS '97)
June 16-19, 1997 Portland, OR
Program Chair: Steve Vinoski, *Hewlett-Packard <vinoski@apollo.hp.com>*
Tutorial Program Chair: Douglas Schmidt, *Washington University <schmidt@cs.wustl.edu>*
Tutorial Submissions due: Feb 6, 1997; Paper Submissions due: Feb 12, 1997

5th Annual Tcl/Tk Workshop '97
Summer '97 TBA
Program Chairs: Joe Konstan, *University of Minnesota,* Brent Welch, *Sun Microsystems*

Application Language Engineering Conference (ALE)
October 15-17, 1997, Santa Barbara, CA
Program Chair: Chris Ramming, *AT&T Research*

11th Systems Administration Conference (LISA '97)
October 26-31, 1997, San Diego, CA
Co-sponsored by USENIX and SAGE, the System Administrators Guild
Program Chairs: Hal Pomeranz, *Netmarket,* Celeste Stokely, *Stokely Consulting*

USENIX Symposium on Internetworking
Fall, 1997, TBA
Program Chair: Carl Staelin, *Hewlett-Packard Laboratories*

7th USENIX Security Symposium
January 26-29, 1998, Marriott Hotel, San Antonio, TX
Sponsored by the USENIX Association, the UNIX and Advanced Computing Systems Professional and Technical Association in cooperation with The CERT Coordination Center
Program Chair: Aviel Rubin, *Bellcore*

For more information about USENIX and its events, access the USENIX Resource Center on the World Wide Web: *http://www.usenix.org*

EDITORIAL

Journalism

My oh my but writing, or magazine sales, (or something) must be challenging. Look at this September's *PC Computing* magazine for some terrific headlines. The top of the cover screams (and I am not making this up): "The New Net Terror WEB VIRUS ALERT Will You Avoid It??" (Yep, that's not just one question mark).

I read the article. People are really concerned with all these various MSWord-borne viruses. Was there a single comment about the Web? Nope. I was crushed.

The rest of the trade press seems to be suffering a bit as well. I read fabulous statements like, "Now that CD quality stereo can be compressed into a 28.8 connection, HDTV video is just a release or two around the corner." I wrote the guy a letter. He pointed me to the company's Web page.

Now why do writers believe all this stuff? I think I'm going to start a campaign of truth in advertising and challenging writers who just go too far. I'll let you know how it goes.

Of course, one must keep one's own yard in order! So, this month we have all sorts of goodies: Our first JAVA column, a nice legal piece by new contributor Nick Nassif, and a potentially controversial article on whether we need to continue with the standard process or whether we should just call it quits now.

I hope you enjoy reading our text. I have not edited out the bald-faced exaggerations and slight flatterings–maybe you can catch them. I feel terrible when I push authors too hard.

Enjoy!

RK

USENIX BOARD OF DIRECTORS

Communicate directly with the USENIX Board of Directors by writing to: *<board@ usenix.org>*.

President:
- Andrew Hume *<andrew@usenix.org>*

Vice President:
- Dan Geer *<geer@usenix.org>*

Secretary:
- Lori Grob *<grob@usenix.org>*

Treasurer:
- Eric Allman *<eric@usenix.org>*

Directors:
- Peter Honeyman *<honey@usenix.org>*
- Greg Rose *<ggr@usenix.org>*
- Margo Seltzer *<margo@usenix.org>*
- Elizabeth Zwicky *<zwicky@usenix.org>*

Executive Director:
- Ellie Young *<ellie@usenix.org>*

CONFERENCES & SYMPOSIA

- Judith F. DesHarnais
  Conference Coordinator
  USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA 92630
  Telephone: 714 588 8649
  FAX: 714 588 9706
  Email: *<conference@usenix.org>*

- Zanna Knight
  Vendor Displays, Marketing
  Email: *<zanna@usenix.org>*

- Daniel V. Klein
  Tutorial Coordinator
  Telephone: 412 421 2332
  Email: *<dvk@usenix.org>*

AUTOMATIC INFORMATION SERVER

To receive information electronically about upcoming USENIX symposia & conferences, finger *<info@usenix.org>* and you will be directed to the catalog which outlines all available information about USENIX services.

PGP INFORMATION

All correspondence to:
Key ID: 969DF939 1996/04/08
USENIX 1996 <pgp@usenix.org>
E5AEECCE8F6963DF 68C5BF4B3CF3FC11

Master key, for signatures only:
Key ID: 2FEA2EF1 1996/04/08
USENIX master key <not-for-mail>
DBA7509966E48AA9 80B2D9E2FEDA005E

USENIX SUPPORTING MEMBERS

ANDATACO
Apunix Computer Services
Crosswind Technologies, Inc.
Frame Technology Corporation
ISG Technology Corporation
Matsushita Electric Industrial Co., Ltd.
Motorola Research & Development
Open Market, Inc.
Shiva Corporation
Sybase, Inc.
Tandem Computers, Inc.
UUNET Technologies, Inc.

More about $IFS

from Chet Ramey
<chet@odin.INS.CWRU.Edu>

Hi, Jerry. This is a response to (and an explanation of) your complaint about the "inconsistency" of `$IFS` in `/bin/sh`. [See *;login:* 21 #4, August 1996]. The problem stems from your confusion of "word splitting," which is one of the shell expansions, and "tokenization," the process of breaking the shell's input into words on which the expansions act.

Tokenization is done by the shell when it parses the input into commands. The words are delimited by space, tab, newline, and the other shell metacharacters ("|", "&", and so on). The result of tokenization is a stream of words and operators. Tokenization does not use IFS at all. This differs from tools like Awk, which uses FS to split the input into fields directly.

After the shell tokenizes and parses the input, if it parses into a command and a set of arguments (rather than, for instance, a while loop), the word expansions are performed on the words. The final expansion performed is word splitting, which is what uses IFS.

I'll annotate some of the commands you used in your *;login:* example with an explanation of the shell's operation.

```
1. _IFS="$IFS"
2. IFS=/
3. HTTP_USER_AGENT="Mosaic 1.2 foo/bar"
4. set/x $HTTP_USER_AGENT
```

After tokenization, there are two words: "`set/x`" and "`$HTTP_USER_AGENT`". The shell expansions work on these two words. Before word splitting, there are still two words: "`set/x`" and "`Mosaic 1.2 foo/bar`". These words are split on IFS, resulting in four words: "`set`", "`x`", "`Mosaic 1.2 foo`", and "`bar`". The "`set`" builtin is called, and the positional parameters are changed:

```
$1 -> x
$2 -> Mosaic 1.2 foo
$3 -> bar
5. echo/$1
```

Before word splitting, this is one word: "`echo/x`". After word splitting, it's two words: "`echo`" and "`x`". Similarly for the next command.

```
6. echo/$2
7. echo/"IFS='$IFS',_IFS='$_IFS'"|cat/-t/-
   v/-e
```

This tokenizes into three tokens, an operator ("`|`") and two words: "`echo/"IFS='$IFS',_IFS='$_IFS'""` and "`cat/-t/-v/-e`". The first word ends up being split into two: "`echo`" and "`IFS='/',_IFS=' \t\n'`", where, as usual, \t and \n represent tab and newline. The second is split into four words: "cat", "-t", "-v", and "-e".

POSIX shells, like `bash` and `ksh93`, use new rules for word splitting: only the words that are the *output* of shell expansions are split. In the first example (line 4), a POSIX shell would try to execute the command "`set/x`". This closes a longstanding shell security hole.

Choosing a Good Password

from David G. Beausang
<dgb@mines.edu>

Dear Editor:

The article "More Secure Mnemonic Passwords: User-Friendly Passwords for Real Humans" in *;login:* volume 21, number 3, June 1996, prompted me to enclose the attached Colorado School of Mines handout titled "Choosing a Good Password."

Some of the examples of good passwords given in the article would fall out of the crack program with the dictionary/definitions we are using. One red flag that went up in my mind concerned the suggested use of common substitutions for letters: 0 for o, ! for i, 4 for a, 3 for e, $ for s, et al.

Please do not consider this to be a severe criticism of the article. It is just that passwords which were good just months ago are now readily cracked.

Choosing a Good Password

[*Copyright by David G. Beausang. Reproduction, with proper credit, is encouraged.*]

First, bad passwords categories.

I. Passwords should never be:

A. Any word in any dictionary, in any language
B. Any formal name or nickname, including spouse's, children's, and pet's
C. Any mythological or fictional character or race
D. Any name of a place (city, country, cross roads, forest, or place of natural beauty), real or fictional
E. Fictional terms
F. Titles of movies, books, compositions
G. The name of any author, composer, musician, actor
H. Any special number
I. Acronyms
J. Phrases
K. Fables or legendary characters or places
L. Combinations of letters or patterns on the keyboard
M. Great license plates you've seen, one2nv, 3vom, ibuy4u, or neat word/letter combinations, aTdHvAaNnKcSe

N. Religious figures, places, or events
O. Anything you can imagine being collected into a list

Examples of bad passwords include: characters and races from Star Trek, the appendices from the Lord of the Rings, pi, e, and the golden ratio, zip codes, THX1138, names of asteroids, names of bacteria, names of viruses, names of algae, names of fungi, names of beers, transliterated words from the Hindu, Chinese, Russian, Yiddish, or any other alphabet, and cartoon characters. A few specific bad passwords include: letmein, youreok, zorkmid, zorro, wonderbread, upchuck, dossucks, qwerty, zaq1234, lmnop, klingon, justforthe, hosannah, hesdeadjim, beammeup. If a password fits in a list, then presume someone has made up that list.

II. Passwords should never be a simple algorithm applied against something in I, above:

A. The "word" backwards
B. Substituting numbers for vowel, r1ch2rd for richard
C. Common substitutions for letters, 3 for e, mov3
D. Appending or prefixing digits, apple639 or 123apple
E. Appending or prefixing special characters, apple@ or $klingon

III. Passwords should not contain information that can be automatically gathered by knowing your user name:

A. Your user name
B. Your user index/number (for UNIX, the UID and GID)
C. User name owner information (for UNIX, the gecos field) which commonly contains your name
D. Information derivable from this information: your initials

This category really is an addition to category I above, but is dynamic depending upon your account information; category I is static.

IV. Passwords should not contain personal information that can be gathered if you are targeted:

A. Your social security number
B. Your student ID number
C. Your phone number, your mother's phone number, your mother's maiden name
D. Your passport number
E. Your street address, the address where you were born
F. Your license plate number
G. Serial number from your camera, computer, stereo

This may seem to be just about everything, right? A good password needs to be something that is not derivable in a semi-automatic manner. The above categories I-III represent known information, or easily derived information, that can be exhaustively applied by a computer to break your password. Category IV represents information that would be applied to specifically break your account, as opposed to any account on a machine. While this may seem to be a very remote possibility, if you are ever personally targeted it is potentially much more damaging to you.

Two final items. Make sure you know how many characters the system allows for a password: a good 15 character password may become a terrible password if the system only uses the first 8 characters. Look at your password selection to make sure it doesn't duplicate a bad password: a (usually) good personal password generation algorithm can generate a bad password; the good and bad may be the result of orthogonal approaches intersecting with a bad password. For example, the potentially good password mxvhall would be bad if your name was Mary Xavier Virginia Hall.

Here are some methods for generating good passwords.

First, if the maximum password length is long enough you can use two unrelated words together, perhaps separated by some punctuation or numbers. For example, parabolasextuplet, peddle$skew, embargo*.umber, pear:xerox, nova::orient. But avoid peanutbutter and lionhunt. Note that if the maximum password length is eight characters embargo*.umber is truncated to embargo* which is easily cracked.

Second, use the first letters of words in a memorable phrase. The phrase "Mary had a little lamb" produces the password mhall. Obviously, memorable is good but traditional or classical is risky. Make up your own phrase. "I got a speeding ticket on 6th avenue" generates igasto6a; "that last calculus exam was real painful" generates tlcewrp.

Third, use grossly misspelled words. For example, fumigayt, lugrnch, phloot.

Fourth, manipulate a good password into a better password: use both upper and lower case characters, add punctuation and/or numbers, depending on what the system allows. For example, igasto6a could become iGAsto6A, mhall could become mHa.*ll$, phloot PHloOT, and MOUTHMOCCASINS MO76UTH81MOC33CASINS.

Five, if you have a good memory, use eight or more, preferably the maximum allowed, random characters.

After you have created a good password, how do you improve the odds of remembering it? Use your new password immediately: change your password and then logout and log back in. After ten minutes (about the length of short term memory) use your new password again: logout and back in. (Changing your password Friday afternoon just before leaving for the weekend can make the new password very difficult to remember.) If you absolutely need to write down your password, make sure that anyone seeing it or finding it can not determine what it is: make sure that it is unrecognizable and can not be associated with your account/user name. This is the same principle that applies to the pin

How often do you need to change your password? The effective half-life of your password depends on its exposure. Piano players can read your keystrokes if they can see your hands. Did you write down your password? (If you had to write it down, the fact it was a necessity does not lower the resultant risk.) Was it accidentally displayed on the screen? Did you login from the hospitality suite at the conference? Or anywhere on a long-haul network? Do you have a nagging feeling you should change it? Is it a good, strong password? It is better to have a good password for months than a bad password for days.

It may seem that you don't have much, if anything, to lose if your password is guessed and your account broken into, but that's not true: you can lose your good name, your reputation. Obscene, racist, threatening email from your account, with your name attached, sent to your friends, family, peers, strangers, and world wide news groups, can be as difficult to overcome and correct as a public scandal.

The USENIX Lifetime Achievement Award

The USENIX Lifetime Achievement Award is to recognize and celebrate singular contributions to the UNIX community in both intellectual achievement and service which are not recognized in any other forum. The award itself is in the form of an original glass sculpture called "the Flame," and in the case of a team based at a single place, a plaque for the team office. The award is presented at the Annual USENIX Technical onference, which in 1997 is to be held in Anaheim, CA, January 6-10.

Past recipients of the USENIX Lifetime Achievement Award are the Computer Science Research Group at the University of California at Berkeley (and a cast of thousands) for the BSD line; Van Jacobson and Mike Lesk for their contributions to networking technology; Tom Truscott, Steve Bellovin and Jim Ellis for their work in creating USENET; and the Software Tools Users Group for popularizing a new vision of operating system software, offering a bridge to portability and power for those limited by proprietary operating systems.

The Software Tools User Group Award

The Software Tools Award recognizes significant contributions to the general community which reflect the spirit and character of those who came together to form the Software Tools User Group (STUG). This is a cash award.

STUG and the Software Tools effort was characterized by two important tenets. The first was an extraordinary focus on building portable, reusable libraries of code shared between multiple applications on wildly disparate systems. The other tenet, shared with the UNIX community, is "renegade empowerment." The Software Tools gave the users the power to improve their environment when their platform provider proved inadequate, even when local management sided with the platform provider.

Therefore, nominees for the Software Tools Award should exhibit one or both of these traits in a conspicuous manner: a contribution to the reusable code-base available to all, or provide a significant enabling technology directly to users in a widely-available form.

The first recipient of this award in 1996 was to Michael Tiemann for the production of GCC, the GNU C Compiler.

HOW TO NOMINATE:

If you believe someone qualifies for either of these awards, we (the nominating committee) welcome your input. Please send us your nomination by December 1 and include your name, details of the achievement, and your individual reasons for making the nomination. Electronic mail to *<awards@usenix.org>* will be fine, or fax to 510 548 5738, or by mail to:

Awards Nomination, USENIX Association, 2560 Ninth Street, Berkeley CA 94710 U.S.A.

Letter from the President

by Andrew Hume
<andrew@usenix.org>

For the last several months, I have been part of a team working on a prototype system processing and analyzing call-related data. There is a *lot* of data involved, but it was brought home to me by a simple `ls` of summary records for one of the data feeds:

```
$ ls -l

total 80523824
-rw-rw-rw- 1 andrew gecko 6627098240 Aug 16 09:23 96144
-rw-rw-rw- 1 andrew gecko 5984369888 Aug 16 09:23 96145
...
-rw-rw-rw- 1 andrew gecko 7191344768 Aug 16 09:24 96149
-rw-rw-rw- 1 andrew gecko 7086484736 Aug 16 09:26 96150
```

These files have been causing me problems, actually; we tend to use local improved versions of things like `sort`, and these all broke on files greater than 4 GB. (The problems were fixed promptly, but do expect some bumps during the migration to 64 bits.)

The sheer size of these files struck me, and I thought back about file sizes during my career. When I started, the standard disk device was a DEC RK05 (4872 512 byte sectors = 2.5 MB), and files rarely exceeded 1 MB. Sizes crept up with improving hardware, but the next size I remember was about 100 MB or so. I first hit these around 1985 when we got our first multiprocessor. My memory is of having to revamp a number of tools, like `cmp`, to handle large files efficiently. `cmp` was a real loser, with an inner loop of get a character from each file and compare it. (Backward compatibility hurt here; being able to report the line number of a byte difference halved the speed of my new `cmp`, so I made it an option.)

Nowadays, I routinely manipulate files in the 5-10 GB range. The odd thing is that I expect the normal tools to work, and by and large, they do so. Although I might complain that it takes five hours to sort a 17 GB file, it is a significant achievement that it "just worked." And I am not the only one: I support a version of `fgrep` that removes the (arbitrary) restriction of 100 patterns that the normal `fgrep` has. A few years back, a user came to me complaining of abnormally slow execution time. I was astonished to find out he was trying to search for about 5 million patterns! (The problem turned out to be quadratic behavior in the kernel for `sbrk`.)

One of UNIX's traits has always been that it runs on the widest range of system sizes of any significant operating system. I am gratified that it can cope with an ever widening range of user needs as well.

USENIX Member Benefits

As a member of the USENIX Association, you receive the following benefits:

- Free subscription to *;login:*, technical features, system administration tips and techniques, international calendar of events, SAGE News, book and software reviews, summaries of sessions at USENIX conferences, Snitch Reports from the USENIX representative and others on various ANSI, IEEE, and ISO standards efforts, and much more.
- Free subscription to *Computing Systems*, the refereed technical quarterly published with The MIT Press.
- Access to papers from the USENIX Conference and Symposia, starting with 1993, via the USENIX Online Library on the World Wide Web *<http://www.usenix.org>*.
- Discounts on registration fees for the annual, multi-topic technical conference, the System Administration conference (LISA), and the various single-topic symposia addressing topics such as object-oriented technologies, security, operating systems, electronic commerce, and mobile computing – as many as seven technical meetings every year.
- Discounts on the purchase of proceedings from USENIX conferences and symposia and other technical publications.
- Discount on BSDI, Inc. products. BSDI information: 800 800 4BSD.
- Discount on the five volume set of 4.4BSD manuals plus CD-ROM published by O'Reilly & Associates, Inc. (800 998 9938) and USENIX.
- Discount on all publications and software from Prime Time Freeware, including Prime Time Freeware for Unix, Prime Time Freeware for AI, Prime Time TeXcetera and Tools & Toys for UnixWare. Contact *<ptf@ptf.com>*.
- Savings (10-20%) on selected titles from McGraw-Hill (212 512 2000), The MIT Press (800 356 0343), Prentice Hall (201 592 2657), John Wiley & Sons (212 850 6789), and O'Reilly & Associates (800 998 9938).
- Special subscription rates to the periodicals *The Linux Journal* (206 527 3385), *UniForum Monthly*, *UniNews*, and the annual *UniForum Open Systems Products Directory (800 255 5620)*.
- The right to vote on matters affecting the Association, its bylaws, election of its directors and officers.
- The right to join SAGE, the System Administrators Guild.

To become a member or receive information regarding your membership status or benefits, please contact *<office@usenix.org>*.
Phone: 510 528 8649.

Reports on the Sixth USENIX Security Symposium

San Jose, California, July 22-25, 1996

Session 2: "The secret whispers of each other's watch"

Summarized by Avi Rubin, Bellcore
<rubin@bellcore.com >

The USENIX crowd initially proved hostile by refusing to laugh at the amazingly funny joke the session chair (and author of this report) told to start the session. However, the undeserving audience was treated to three very good papers. The papers were especially relevant given the focus of the conference on the applications of cryptography.

The first paper was titled "SSH – Secure Login Connections Over the Internet". There was tremendous interest in this report because SSH seems to have avoided the usual IETF battles and become a de facto standard because it is simple, useful, freely available, and easy to use. There was even a flyer at the vendor displays for an SSH product. The talk was given by the author, Tatu Ylönen, of SSH Communications Security, Ltd., Finland.

SSH provides a secure transport layer supplying authentication, integrity, and confidentiality to users. It is intended to replace existing commands such as `rsh`, `rlogin`, `rcp`, `rdist`, and `telnet`. There is direct support in SSH for SOCKS and FWTK. In addition, there is a mechanism in SSH to forward the connection to a user's X server over the secure channel. SSH uses 1024 bit RSA keys for user authentication. Almost any platform you can think of is supported, except for the Macintosh version, but a Mac version is expected out this fall. Performance does not seem to be impacted significantly by SSH.

The packages are freely available at *<http://www.cs.hut.fi/ssh>*. This page is accessed about 1,000-2,000 times/day. Access has come from over 6,000 different hosts.

People seemed eager to get their hands on the code. The only question from the audience was about the availability of a Windows version. (It is available.)

The second talk, by Barry Jaspan, was titled "Dual-Workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks." He addressed the problem that many systems, such as Kerberos, use the current password to protect future passwords. In such a system, if the attacker ever discovers a user's password, he will be able to determine all future passwords by simply observing network traffic. To solve this problem, Jaspan proposed to integrate the encrypted key exchange protocol of Bellovin and Merrit into Kerberos.

The problem with this idea is that it requires an expensive exponentiation operation that can be very slow. Therefore, in the new protocol, a shorter than normal modulus is employed, but it is not used to encrypt any known plaintext. Since the plaintext is not known, there is no way an exhaustive search can be used to discover the Diffie-Hellman key. Jaspan suggested that an attacker would have to perform a discreet logarithm for every password guessed; thus, if the search space for passwords is large, the discreet log problem does not have to be intractable for any given instance, but for a combination of a large number of instances.

Jaspan pointed out that his protocol is safe only against passive attacks. Active person-in-the-middle attacks are much more difficult.

An audience member brought up a valid point. If the user has any information at all about the types of passwords that a user picks, the workfactor is not necessarily the multiple of the two schemes. In fact, this idea probably needs to be explored further before such a protocol is deployed.

The last paper in this session was titled "Security Mechanism Independence in ONC RPC." The authors were Mike Eisler, Roland Schemers, and Raj Srinivasan of SunSoft, Inc.. Mike Eisler presented this paper which proposed a mechanism for layering ONC RPC on top of the GSSAPI. The goal is to transparently provide security services to RPC calls while maintaining compatibility with systems such as Kerberos 5. The authors provide an interface to the programmer with a new security flavor, RPCSEC_GSS. The paper describes how to create contexts between clients and servers with different levels of security. Performance numbers seemed to indicate that, as expected, integrity does not cost that much, but that confidentiality can be quite expensive. An internet draft has been submitted to the IETF to standardize on the ONC RPC protocol.

Session 3: "There are more things in heaven and earth"

Summarized by Mark K. Mellis
<mkm@mellis.com>

Establishing Identity Without Certification Authorities

Carl Ellison, Cybercash, Inc.

Carl discussed the limited utility of the conventional notion of identity certificates when applied to the problem of confirming identity in a human context – that of establishing a

secure channel of communication with an old friend with whom you've lost touch.

He explored the concept of identity as the binding of a nickname to a body of shared experience and memories, and as the binding of a public/private key pair to a distinguished name. This led to the question, "How does one map a nickname (your handle for the old friend) to a distinguished name (with a well-known public key) without a secure channel over which to verify the shared experiences?"

On the way to the old friend, out of touch protocol, Carl took us through definitions of identity, contrasting types of identity certificates, name spaces, mechanisms for exchanging public keys, and vulnerabilities in those mechanisms. The balance of the presentation detailed a protocol for two "old friends" to exchange public keys and then determine whether or not their communications channel had been compromised by a person-in-the-middle attack.

Using an interlock protocol to ensure that answers come from the directly adjacent neighbor, be it the old friend or the person-in-the-middle, rounds of questions about shared experience are exchanged and the entropy of the received answers is computed. Based upon the computed entropy, the participants decide whether the neighbor is their old friend or a stranger, and the protocol ends.

Secure Deletion of Data from Magnetic and Solid State Memory

Peter Gutmann, Department of Computer Science, University of Auckland

Peter's presentation was one of the rare USENIX talks that draw on the material world, that of magnetic fields on disks. With the aid of a number of beautiful graphics, Peter described techniques for recovering "erased" data from various types of disk drive using Magnetic Force Microscopy (MFM). He discussed data persistence in semiconductor memory in his paper but didn't cover it during his talk due to time constraints.

MFM works by moving a sharp magnetic tip over a surface and using optical interferometry to measure the deflection of the tip caused by stray fields. Peter asserts that using this technique, even a relatively inexperienced operator can begin to recover data from a disk platter within ten minutes. Using more sophisticated magnetic force microscopes, data gathering could be automated and entire disk platters could be imaged. He displayed images that clearly showed how erased and overwritten data can be recovered. Examples of this imagery are available at:
<http://www.di.com/Theater/nt_mfm.html>

Peter suggested measures that can be taken to prevent critical information such as cryptographic keys from being paged out to disk as the result of normal virtual memory system operation, since once on disk the data is vulnerable to compromise. Another technique for data protection involves the use of the smallest, highest density disk technology, since that makes data recovery more difficult.

During the question and answer period an inquiry was made about government standards for removing data from disks in a secure manner. Peter responded that all such standards were classified. He also commented that the best way to protect data on magnetic disk was to melt the disk down to a "pile of molten slag," prompting Steve Bellovin to suggest that systems be equipped with "`/dev/thermite`".

A Revocable Backup System

Dan Boneh and Richard J. Lipton, Department of Computer Science, Princeton University

Dan described a strategy for rendering data stored off-line unusable. This is of particular interest to the system administrator who is required to implement a policy where, for example, backups of email directories are only retained for one week while all other data is retained for six months. Using traditional methods might require two separate sets of backup tapes, one for email and the other for the balance of the data, whereas with this system a single set of backups could be performed and the system told to "forget" the email data at one week intervals.

The system is built around a hierarchy of keys that are themselves encrypted. All data written to tape is encrypted using a block cipher. The keys are stored on-line, with the exception of the master key. The master key is used to encrypt the per-file keys, and is kept off-line. A data set is "forgotten" by deleting the key which was used to encrypt it. By "forgetting" data sets as a routine business practice, legal issues focused on opportunistic destruction of corporate records are avoided.

This system has been implemented via a modified version of the dump command that uses a triple DES block cipher to encrypt the data stream. Files are recovered from tape using a wrapper around the standard restore command. A simple user interface has been constructed. Little slowdown in backup and restore has been observed.

During the question and answer period Dan commented that although the content of individual files was encrypted, file system meta-data was not encrypted.

Session 5: "In the middle of her Web, which spreadeth wide"

Summarized by Avi Rubin, Bellcore

The first paper in this session was titled "Chrg-http: A Tool for Micropayments on the World Wide Web" by Lei Tang and Steven Low, AT&T Research. This paper provides a billing mechanism within an intranet that uses Kerberos. The idea is that rather than implement one of the heavy-duty payment schemes, such as SSL, iKP, or SET, sites that already have Kerberos installed can add a new method to their Web server and use the security mechanisms within Kerberos to authenticate transactions. The idea is not to provide an Internet-wide solution, but rather to enable Kerberos administrators to enable micro-payment charges within their domain. The implementation used NCSA Mosaic and Kerberos version 5. There was no performance data available because it was a three-month summer project in 1994 at AT&T Bell Labs. It does not appear that there is any code available.

The second paper was presented by Trent Jaeger, winner of the best student paper award for this work. The title was "Building Systems That Flexibly Control Downloaded Executable Content". The other authors were Avi Rubin and Atul Prakash. This paper addressed some of the access control issues with regard to remote executables that are downloaded and run locally. The paper presented an architecture that flexibly controls the access rights of downloaded content. Thus, the writer of an object can specify what the reader can and cannot do to the object. Similarly, the reader of the object can specify what the object can and cannot do in the local environment.

An access control model was presented where objects control access of principals to perform various operations, and a specification model is given for developers to specify the access rights of principals in their applications and how rights can be transformed given user actions. This paper tied in nicely with the first paper of the conference, which dealt with restricting the access of helper applications to the local environment.

The final paper in this session was "Enclaves: Enabling Secure Collaboration Over the Internet". It was presented by the author, Li Gong. This was one of those few papers presented at a conference that makes you say to yourself, "Boy, if I could have that right now, it would change the way I do things." Gong presented his enclaves system, which allows groups of users to set up private, virtual subnets. This is done without requiring any firewalls or heavy duty transport layer security.

The enclaves API sits on top of TCP/IP and has been implemented on several different platforms. Three applications were presented: a secure whiteboard, which uses group multicast; a virtual shared filesystem; and a collaborative editor. The latter works by allowing members of the group the ability to lock a particular section of text while they are working on it, and all locks are displayed to all members of the group.

The group protocol consists of a group leader and group members. Serialization of messages is achieved by having everything go through the leader. The main idea behind enclaves is that they allow users to control which services their machines provide and enables users to communicate only in secure mode, if they so choose. The plan is to release a version of this soon. It is sure to be well received.

Uniforum Panel Sessions

Summary from Uniforum's Uninews

The Security Symposium was cosponsored by UniForum and with the cooperation of the Computer Emergency Response Team (CERT). Over 800 security experts, system administrators and analysts, site managers, and programmers attended, almost doubling the attendance at the 1995 symposium in Salt Lake City.

A one-day track of the symposium was UniForum's four panel sessions: Security and Privacy Issues; Electronic Commerce; Cryptography and the Law; and Cryptographic Infrastructure.

Something for Everyone

The first UniForum panel session was Security and Privacy Issues, chaired by Peter Neumann of SRI International. The panelists were Mary Connors of Computer Professionals for Social Responsibility, Jose Martinez of Sausalito Associates International, and Gio Wederhold of Stanford University. They concentrated on outlining and defining what the essential privacy questions are, and how they can be addressed and solved. Among these issues were access to information contained in government systems – such as those of systems of the Internal Revenue Service, the Social Security Administration, and state departments of motor vehicles; private companies' practices involving the use of lists of names; and the troublesome fact that if you are networked in any way, you can never be sure that *anything* is completely secure.

Gio Wederhold focused his presentation on the issue of medical records security, which is an area of growing concern. The major insurance companies have virtually total access to medical records, while the subjects of those records – patients and other medical consumers – have limited control of their own information, and that includes the results of medical research. From this scenario, Wederhold transitioned to the concept of the "virtual company" and how its

development is being seriously hindered by the fact that current practices make it difficult for people to trust each other.

Mary Connors rounded out the session with a history of the development of major privacy legislation in both the federal government and the California state government, citing the Freedom of Information Act, the Privacy Act, and the California constitution, with its right-to-privacy provision.

At the panel on electronic commerce, session chair Rik Farrow started with a question: "How many of you are currently doing some kind of e-commerce?" About half the people in the room raised their hands, showing the pervasiveness of this activity. Farrow pointed out that even now many companies want to get into electronic commerce, but confess to fears about how to do it securely. He went on to say that e-commerce has taken business to a level of abstraction never before seen. The object of the transaction is not directly experienced by the buyer before sale; the Web site being used may or may not belong to the entity the buyer thinks he or she is dealing with; and to top it off, the funds used to pay for the transaction are electronic.

The panelists – Fred Avolio of Trusted Information Systems (TIS), Daniel Geer of Open Market, and Bruce Schneier of Counterpane Systems – discussed a variety of topics within electronic commerce, including concerns about authorization, accountability, integrity, and confidentiality, and what the world might look like when electronic commerce is the norm rather than the exception, including a prediction of the "disintermediation" or the declining role of intermediaries in commerce.

Lively Discussion

The third session, Cryptography and the Law, was by far the liveliest and most contentious of the panels. Session chair Dan Appelman of the law firm of Heller, Ehrman, White & McAuliffe assembled a panel consisting of John Gilmore of the Electronic Frontier Foundation, and, through a teleconference with the office of US Senator Conrad Burns, three members of the senator's staff and Scott Charney, chairman of the computer crime unit of the Justice Department.

Sides were taken immediately, with only Charney defending the use of key escrow policies and limitations on encryption technology. His point of view (which he defended valiantly in the face of strong opposition from the others) was that "robust" encryption can and should be available, but that legitimate national security and public safety concerns warrant restrictions on encryption. Those on the panel in favor of unrestricted cryptography pointed out that, because of the truly international nature of the Internet, laws passed by any given country limiting electronic security technology are largely "irrelevant" and cited the lack of effect that France's key escrow policy (the only one in the industrialized world) has had beyond its own borders.

The final panel session, Cryptographic Infrastructure, was chaired by Fred Avolio of TIS. The panelists were Peter Dinsmore, also of TIS; Carl Ellison of Cybercash; Constantin Tanno of Morgan Stanley & Co.; and Walter Tuvell of the Open Software Foundation. This session covered such topics as architectures for public key infrastructures; key recovery and backup; and public key identification, authentication and authorization.

Invited Talks

Just another convicted Perl hacker

Randal Schwartz, Stonehenge Consulting Services
Summarized by Tina Darmohray
<tmd@usenix.org>

Randal Schwartz discussed the events leading up to, and the personal nightmare following, his conviction in the case of the *State of Oregon v. Randal Schwartz,* Washington County Circuit Court C94-0322CR. The complaint was brought by Mr. Schwartz's client, the Intel Corporation. He gave insight into the thoughts and logic behind his actions that led to Intel's complaint in November 1993 and covered the details of his legal experience and the Oregon laws under which he was convicted.

His goal in sharing his story was to educate other computer professionals in order to prevent what happened to him from happening to them. Additionally, Randal urged computer-literate professionals to become aware of laws that can affect them personally or professionally and to become involved in creating and modifying computer-related law to make it reality, based and fair.

Updates surrounding the status of Randal's appeal and what you can do to get involved are kept on *<http://www.lightlink.com/fors/>*.

Notes from the CERT and TIS Firewall Toolkit BoFs

by Christine Hogan

CERT BoF

The CERT BoF focused primarily on a discussion of the trends the CERT team are currently seeing in reported incidents and the changes that are happening in CERT.

Current trends in the incidents reported to CERT are more root compromises and more sophisticated attacks, often by unskilled people who have access to kits. One cited example

involved root being compromised, followed by the perpetrator attempting to execute DOS commands. Between 5 and 15 incidents are reported per day, most involving exploits of bugs that have previously been addressed by advisories. The most popular are `sendmail`, `rdist`, `telnetd`, and `httpd` vulnerabilities.

Sniffing attacks are also very common, particularly on Linux systems. IP spoofing attacks, also prevalent, are generally not spotted by the victim site, but by another site that informs the victim administrators. Port and IP address sweeping also happens, but few sites appear to detect these probes.

Someone asked whether commercial firewalls are being broken. Apparently they are not, but one member of the audience had seen a compromise through a commercial firewall that was misconfigured.

The main messages from this portion of the BoF were, first, even large backbone ISPs have been victims of sniffing attack: thus any data you send off your network can be compromised. In addition, vigilance in applying patches and monitoring is key. Not enough people are monitoring for probes.

On the topic of the evolving role of CERT, the team explained that since DARPA has changed its funding policies, CERT has needed to move away from incident response and toward research based on the statistics they have been gathering since 1988. A number of commercial entities now provide services that include incident response teams to which a company can subscribe. CERT has also gone into this business, though response to life-threatening situations or to attacks on the infrastructure of the Internet will still be the highest priority.

CERT intends to continue in its role as a central point of contact that gathers information for statistics, trend recognition, and spotting when an incident is part of a larger attack. The team has been talking to the other companies that are providing incident response services about getting statistical information from them, and thus far the response has been favorable. The team also encourage ISPs who are brought in by their customers under confidentiality agreements to deal with an incident to work with the customers to provide suitably sanitized information to CERT so that it can gather statistics and recognize when the victims are part of a larger incident.

Finally, the CERT team encouraged everybody to use the form available by FTP from *info.cert.org* when reporting incidents to CERT, and to report incidents even when the "Action Requested" is just "FYI" or "Pass this on to others who are affected." Also, on prompting from Brent Chapman, they announced that they have finally put up a Web site on *www.cert.org,* which they promised to get around to populating with more information in the next 30 or 40 years.

TIS Firewall Toolkit BoF

As usual, the majority of the discussion at the TIS `fwtk` BoF centered around when the next release of the toolkit was likely to happen, and what TIS could do to help improve the process of making available patches that have been contributed by the user community. This latter was accompanied by a discussion of implied endorsement of those patches by TIS.

At the recent BoF at SANS in Washington, DC, in May, Fred Avolio (who was also hosting this BoF) had said he believed the next release would happen in July. Unfortunately, this date had to be changed to August because of the impending release of Gauntlet 3.2. He made a commitment to the group, however, that there would be at least one new release per year of the toolkit and said he hopes it will be slightly more frequent.

On the issue of patches and feature enhancements contributed by others, the provision of a place on their FTP server for unreviewed, contributed patches was discussed as a likely solution. This directory would be subject to a specific disclaimer that the code had not been in any way reviewed or endorsed by TIS, because the reviewing and approving process is the bottleneck. TIS will rely on the user community to be vigilant and inform them of problems with the contributed code.

Some feature enhancements were also discussed, one of which was firewall-to-firewall encryption. Fred replied that this really needs to be part of the IP stack, which requires kernel modifications and thus is unlikely ever to appear in the toolkit. A somewhat related feature suggestion was an encrypting `plug-gw`. Someone in the audience has already done the work on this and was willing to share it. That person was encouraged to send code to TIS, which could make it available in the contrib area. The issue of making it available outside the US was raised, because the developers are not from the US, and Fred promised to check with their lawyers to see if they can legitimately put a pointer to where it is available on their FTP site. The work that the Linux community is doing on porting the toolkit was also discussed, and it is being merged back into the main source tree.

Fred also solicited help from the community in a number of areas. First, the documentation needs a complete overhaul, and better documentation than that provided by TIS has been written by people in the user community. Those who have done work in this area, or who are willing to do so, were encouraged to contact him by email at *<avolio@tis.com>*. In addition, TIS would like to have some people who do not work there on the `fwtk`-support mailing list. If you are willing and able to help support the toolkit, contact Fred. There

was also a suggestion that the `fwtk` users list should be moderated to improve the signal-to-noise ratio. Fred felt that moderator should not be a TIS employee, but he also stated that the TIS folks would still like to see all mail that was sent to the list, whether approved or rejected. If you would like to volunteer to moderate the list, drop him an email.

Finally, the attendees of the BoF expressed their appreciation of all the work that TIS has done and continues to do in order to make the toolkit available and support it.

Musings: The Sixth Security Symposium

by Rik Farrow
<rik@spirit.com>

As I write this, it's 100 degrees outside (38 Celsius). There are a few fluffy clouds and a rather threatening looking thunderhead off in the distance. It's hard to imagine what the weather will be like when and where you read this. It's even harder to write about what happened in the past for someone reading in the future.

The week of July 22 marked the occurrence of the Sixth USENIX Security Symposium in San Jose, CA. Expectations were high, and in the main I was not disappointed. Although the fireworks were muted in mostly polite discussions, I'll attempt to fill in those of you who could not attend yet might be interested.

To the surprise of many of us, BoFs started on Tuesday night, the night before the opening of the symposium. I did manage to attend the Firewalls BoF, although I missed the beginning of it and all of the CERT BoF. Having attended past CERT BoFs, I can surmise a good portion of what went on: Ed DeHart explains what CERT is and is perhaps a little defensive when someone inevitably complains that CERT didn't do enough for his or her organization. Then DeHart points out that network invaders, a.k.a. hackers, are using old holes to break in and gain root privileges. CERT has estimated that attackers gain root in about 20% of all incidences.

What's new is "an increasing sophistication in techniques," to paraphrase a CERT summary. Attackers use tools to analyze the pattern of system calls made, looking for sequences of calls that have proven to be flaws in past programs, for example, to run a race to create a symbolic link (localmail). Source code reading has become fashionable, with Linux becoming a common victim. There are even new twists on old problems, such as a new way to abuse the expreserve program.

Marcus Ranum made a valiant effort earlier in the day to head off these problems. Ranum's new course covers secure programming techniques in an attempt to educate programmers in avoiding those very coding techniques that attackers love to exploit. Ranum's course also included a section on encryption, something we will all need to learn more about in the future, and the theme of the conference.

Steve Lodin posted his notes for the Firewalls BoF to the Firewalls mailing list, so I won't repeat them here. Those notes, along with over a megabyte of other Firewalls postings, can be found at *<ftp://ftp.greatcircle.com/>*, in the archive for August 1996.

Parsed Strings

Ron Rivest, a professor at MIT's Lab for Computer Science, led off the conference with a keynote describing Simple Distributed Security Infrastructure (SDSI). You can get the draft paper for SDSI, pronounced "sudsy" (a real improvement over SCSI), at *<http://theory.lcs.mit.edu/~rivest>*. For those of you who just want some idea of what this is, read on.

Rivest began by praising the inventors of public key encryption, Whit Diffie and Scott Merkle. This "marvelous insight" makes it possible to send encrypted data using the recipient's public key or to digitally sign your own data using your private key, allowing those with access to your public key to verify that you have signed the data they received.

I should mention that Rivest is the "R" in RSA, and the designer of RC2, RC4, and MD5. Patents controlling public key-private key technology have long been an impediment to their use in the US, but not elsewhere, as many countries do not permit patents to be issued for innovations that entered the public domain through papers. The US patents will shortly expire, perhaps changing the scenery. But that is another story.

Rivest's main thrust surrounds the problem with obtaining and managing public keys. PGP, for example, relies on informal networks for distribution of keys. PEM relies on a more formal certification agency for obtaining public keys. Neither scheme scales very well, in that it is currently impossible to obtain the key of someone you have never met but might know the email address for.

SDSI describes a formal key distribution system with some of the features of DNS–that is, a distributed hierarchy with a local namespace. SDSI syntax defines S-expressions, which is where the parsed strings show up. Rivest's and Butler Lampson's implementation is very reminiscent of LISP, using nested parentheses to indicate expressions and subexpressions. Rivest claimed that one real advantage of SDSI is the ability to define groups, that is, multiple principles (keys that represent an individual or organization) collected into a group entity.

I thought the most interesting concept was the notion of relative naming. For example, to specify a person named bob who is known to ron in a certificate, you could write (ron's bob) or (ref: ron bob). In a way, this is very similar to the Internet email addresses we use, for example, *<geer@usenix.org>* means that geer is known at USENIX, or (usenix's geer) in SDSI terms.

SDSI certificates contain expiration dates and will be signed by other principals. One basic orientation of SDSI is that servers will always be available, that is, online. For example, you could get a membership certificate and present it when you want to access some resource available to that membership. The server, in turn, would need to access the owner of that membership to check the principal used to sign your membership certificate.

In the question-and-answer period, Peter Honeyman asked what would happen if the server won't disclose the ACLs (which might permit you access if you knew the right principal) and the client refuses to disclose all principals held. This appeared to me as a new and intriguing form of deadlock. Rivest answered by saying that you could do anything you wanted with SDSI. Matt Blaze described SDSI as a "powerful concept," but had some problems with the naming mechanism and announced a public key infrastructure BoF later that evening. Someone else asked where the initial keys come from, to which Rivest answered that it was a bootstrapping problem.

There are other proposals for solving the public key problem, including the IETF's Simple Public Key Infrastructure (Carl Ellison, of CyberCash) and Blaze, Feigenbaum, and Levy's decentralized control model (try *<ftp://ftp.research.att.com/dist/mab/policymaker.ps>*).

Matt Blaze, Carl Ellison, and Ron Rivest all showed up after the reception to politely discuss problems with SDSI. In brief, Blaze pointed out that most public key schemes focus on email, but are being used (or abused) to support other uses that they do not fit. Not included in SDSI is the notion of delegation (third party's use of keys) or capabilities (that a principal be assigned to a particular use only, its capability). I particularly liked it when Matt compared X.509 key certificates to the PGP scheme as the "fascist vs. the people" techniques.

There's more, but this is not my specialty, and I'm sure I'll hear about any misconceptions I may have spread.

Secure Environments

The best paper this year was a student paper. Ian Goldberg, David Wagner, Randi Thomas, and Eric Brewer of the University of California, Berkeley, won the honor (and a check) for their "Secure Environments for Untrusted Helper Applications". Their idea was to filter system calls made by applications loaded from untrusted networks, and also calls made by helper applications (which may be too complex to be trusted). Their solution, named Janus, captures selected system calls using the `/proc` interface used in Solaris. The advantages of this system are speed (low overhead), flexibility, reliability, and the fact that the modifications all exist in user space. The main disadvantages currently are that the work is not very portable and requires `/proc` to work.

I liked the concept–very clean, conceptually simple, and a working implementation (try *<http://www.cs.berkeley.edu/~daw/janus/>*). When I told this to Ian Goldberg, he said he was surprised at the fuss about something so simple.

Bill Cheswick presented a paper that he and Steve Bellovin and he wrote about a DNS filter and switch. The concept, again, is quite simple. Build a proxylike program that receives DNS requests and answers, filters out unusual or unrequested answers, and passes the results to the appropriate, real, DNS server. Bill mentioned that he and Steve had seen packet injectors and related tools in hacker toolkits (used to add spurious glue records to DNS caches). Unfortunately, "Dnsproxy," caught in the web of lawyers and multiple corporate policies, is not currently available for downloading.

Legal Encryption

Dan Appelman chaired a controversial panel Wednesday afternoon. Only Dan and John Gilmore of the Electronic Freedom Foundation were actually seated in front of the crowded room. The other panelists were teleconferenced, with some minor technical difficulties (making the discussions essentially simplex). Several aides to Senator Burns (Mike Ross, Brett Scott, and Matt Raymond) worked as one remote team. Senator Burns is the sponsor of pending legislation that would remove most restrictions on encryption. If I remember correctly, the thrust of this bill would be to provide a level playing field for US software and hardware vendors, making it possible to export from the US any technology that can be imported from other countries.

If this doesn't make sense, try this one. A number of years ago, UUNET had designed a network device that would encrypt the payload of TCP IP packets based on the destination IP address. This device, a "blue box," used a German-manufactured DES chip for encryption. Even though the chip was obviously imported, the blue box could not be exported because it included the chip. The US Commerce Department is responsible for issuing licenses for exporting munitions, such as encryption, under current US law.

Scott Charney represented the US Justice Department and the opposing side of the panel (all by himself). I first heard Scott Chaney speak during the Third Security Symposium in Baltimore, back in 1992 (when George Bush was president). I enjoyed Charney's practical advice back then, but had a harder time listening to him this time. Essentially, the Department of Justice, the FBI, and other agencies are adamantly opposed to strong encryption appearing in mass-market products, for example, Microsoft Windows. Charney stated that investigators had already encountered encrypted data which made investigations difficult to pursue. When asked how many times this had occurred, the answer was vague (my impression was less than ten in the last year).

The current administration, like the one before it, agrees with the Department of Justice. Even though we can easily acquire strong encryption, such as PGP, the US Government intends to make it difficult for software vendors in the US to distribute products that incorporate strong encryption–unless the US Government has access to the encryption keys (witness Skipjack and the Lotus Notes agreement). So, while the Chinese ship container loads of AK47s into San Francisco, and the amount of heroin annually imported into the US is measured in the tens of tons, encryption software will remain restricted, unless we buy it or acquire it someplace else. (Note this is a problem for many countries other than the US, for similar reasons.)

Although I sympathize with the agencies, I think they are trying to command the tide to turn back.

More ITs

Randal Schwartz, well known for his Perl books, spoke in front of another packed house. Schwartz was convicted of three felonies for nothing more than overzealous system administration. In other words, Schwartz's only motivation for what he did is that "doing his job well might advance his career." I had first heard Schwartz speak about a year ago, and his experience has deeply affected whom I am willing to work for and the contractual conditions for that work.

I also caught Marcus Ranum's IT about firewalls. "Firewalls are the padded dashboard phase of network security," said Ranum, an allusion to older cars (before seat belts). Given the increasing complexity of desktop applications, desktops must become security aware. Ranum said that perhaps IPv6, with its built-in encryption and digital signature capabilities, might help. But will IPv6 happen soon enough? Meanwhile, we have new services, such as Java and ActiveX, plus great new features, such as embedded URLs in RealAudio in the next release.

Ranum has embarked on a search for return-on-investment. If firewalls and other security technology are to be sold successfully, you must be able to prove that the risk of not making the investment is less than the cost of the investment in security technology. Ranum's back-of-envelope calculations showed him that the average IT group spends less than $50k per year for security, the average incident for a company at risk costs about $100k, and the average IT security product costs $200k. In other words, companies are not paying as much as their risk per incident per year, but if they spent the $200k, their return-on-investment would be negative.

In the long term, Ranum spoke of three solutions. First, train software engineers in secure programming techniques so they can write good code the first time. Alternatively, make release 2 secure. Second, provide high-quality support tools and secure versions of libraries. Third, build security into the infrastructure: vendors, standards, and end users.

The current prognosis: nobody is ready to fix things. "Padded dashboard tastes good," quipped Ranum. Although this might seem to be ending on a negative note, you can bet that Ranum, and others like him, will be working on creating secure solutions–solutions without hidden problems that will bite their authors years later.

Report on the Fourth Annual Tcl/Tk Workshop

by Mark Roseman
<roseman@cpsc.ucalgary.ca>
and Mark Diekhans
<mark@grizzly.com>

USENIX sponsored the fourth annual Tcl/Tk workshop on July 10-13, 1996 in Monterey, CA. It brought together an eclectic set of 162 or so Tcl/Tk practitioners for several exhausting days of delving into new tools, applications, and techniques. The workshop reflected the exciting, fast-paced innovation that is happening throughout the computing community.

For those unfamiliar, Tcl (tool command language) is an easy-to-learn scripting language, while Tk is its companion user interface toolkit. They were created by John Ousterhout while he was at the University of California at Berkeley, and the core development is now hosted at Sun Microsystems Labs. Beginning life as a UNIX tool, Tcl/Tk now also runs on Macintosh and Windows. Tk is also a popular GUI toolkit for languages like Perl and Python. It is estimated that the Tcl/Tk developer community is 300,000-500,000 strong.

It isn't a surprise that the Internet was one of the big themes at the workshop. The big news was the announcement of the Tcl/Tk plug-in from Jacob Levy at Sun Labs, which allows scripts or applets written in Safe-Tcl ("tcllets") to be run inside Netscape Navigator. This was heralded as a great delivery platform by the attendees, and also as a great improvement on current Web programming (Tcl/Tk scripts are normally far shorter and easier to write than a corresponding Java applet). To complement the plug-in, Tcl/Tk was the feature article on the popular Sun WWW home page. This generated an enormous amount of interest, as well as saturating the network connection to Sun Labs' FTP site, as thousands rushed to download the plug-in.

With all of the hype about the World Wide Web, those of us in the Tcl/Tk community were not about to be left out. With two related sessions, one on WWW applications and one on tools, we had a lot more to offer beyond the famous plug-in.

The first paper of the workshop, on the TeamRooms groupware application, showed an exciting use of Tcl/Tk in building an ambitious collaborative system. It turns the Internet into a true productivity tool and supports realtime interaction with a workgroup. Who says the Internet is just a productivity drain? Another paper presented an application of the Tcl language to Internet programming called Agent Tcl, a powerful system that sends agents off to other systems to do various tasks. To silence those who falsely believe that Tcl and Java are competing languages, a paper was given on TclJava, a set of classes that allow extensions in Tcl to be written in Java instead of C or C++. The end result is not only a powerful scripting language to use with Java, but also portable extensions for Tcl. We also got to see Tcl and Tk used to develop innovative versions of some of our favorite Internet applications. The NR Newsreader, with its GroupLens filtering system, can actually help one avoid "make money fast" postings. Another paper from "down-under" described SurfIt!, a WWW browser written in Tcl and Tk and the world of of Tcl applets and active messages. The two most useful papers for those of us who have to develop WWW applications dealt with Tcl/Tk HTML Tools (for parsing and displaying HTML), and the power and flexibility of using Tcl as language to develop CGI scripts.

A standard feature of these workshops is John Ousterhout's "state of the union address," which reports on the progress of Tcl/Tk over the past year. This included the first Mac and Windows versions, dynamic loading, sockets, Safe Tcl and the plug-in, and the first version of the SpecTcl user interface builder (and its "evil twin" SpecJava). The session also provides a discussion forum where the developer community is invited to help set priorities for the next year. As always, a lively–if mathematically suspect–set of votes ensued. Two big developments are expected before the next workshop. The first is a version of Tk that will support native widgets on Macintosh and Windows platforms (the current implementation uses an X11 emulation layer), along with corresponding changes to menus, fonts, events, and so on.

One of the most frequently demanded enhancements is to speed up scripts. Brian Lewis' presentation about the on-the-fly byte-code compiler Tcl being developed at Sun Labs gave us hope that a solution is around the corner. For those who can't wait, we got an update on the ICEM CFD compiler that was presented at last year's workshop.

Scientific programming is generally thought to be the realm of modern, high performance compiled languages, such as FORTRAN. However, the papers presented in the Scientific Applications session demonstrated that Tcl and Tk can be valuable tools in developing such applications. The session started off with the winner of this year's best paper award, "Lesson from the Neighborhood Viewer: Building Innovative Collaborative Applications in Tcl and Tk," presented by Alex Safonov and Douglas Perrin. The program is a collaborative browser for image databases. It was constructed specifically for viewing brain images. In this presentation we not only learned about the process of developing such an application, but also about many related issues, including neuroanatomy, constraint programming, and wavelet compression. In the next paper we explored the world of high energy physics with a high performance graphic display application from the European Synchrotron Radiation Facility. The next paper dealt with building image and volume visualization tools using Tcl and Tk. This included a peak at a new widget for grey-scale images. The final paper brought us back to neurology, with a patient information system for The Cleveland Clinic Foundation. This included dealing with the ugly, but persistent issue of an installed base of character terminals. Many people came away from this session feeling that Tcl and Tk were ideal tools for quickly building interactive scientific applications.

Anyone who has embedded Tcl into an existing application realizes that one of the most time consuming tasks is developing the interface between Tcl and C or C++. Two papers offer hope that this rather dual task will become almost nonexistent: SWIG is a tool to generate wrapper commands for C and C++ function for not only Tcl, but also Python and Perl. Another paper from GE gave C++ programmers an automated method of wrapping Tcl commands around their classes.

There were a number of other innovative systems presented. Here is just a sampling: Robert Gray described Agent Tcl, a sophisticated framework for migratory agents built on Safe Tcl;Steve Grady talked about a database test system at Sybase that including the test cases contains over 1.5 million lines of Tcl code! Dave Griffin described how Tcl is used inside Forum, a groupware discussion product from DEC

Alta Vista; Jason Stewart gave an update on Pad++, an absolutely incredible zoomable interface system that must be seen to be believed.

Just prior to the workshop, Jacob Levy and Tom Christiansen organized a smaller, by invitation, workshop to discuss Tcl/Tk interoperability issues. It included participants from the Perl, Python, and Scheme communities who have been using Tk as their GUI toolkit. The general consensus was that the upcoming Tcl compiler would help with a number of the multiple language issues. It succeeded in getting the different groups together and helped to foster a greater sense of cooperation. Some of the other issues tackled in this workshop were support for threads, I18N, and namespaces.

There was also a plethora of t-shirts at this year's workshop. Besides the "official" workshop t-shirts (thanks to Don Libes and USENIX for producing these), Sun Microsystems unveiled the official Tcl/Tk logo (a stylized feather, Tcl being pronounced "tickle") by giving out shirts to everyone. Michael McLennan and crew from AT&T (who develop fine utilities such as [incr Tcl] and BLT, as well as offering Tcl/Tk training) gave away no less than four varieties of t-shirts, as well as basketball hoops and yo-yos. We've hit the mainstream!

We also reflected upon how the efforts of a lone professor hacking away in the dungeons of UC Berkeley had led to such an incredible community of developers that has created a very powerful set of tools to solve many of our programming problems. We all await the excitement of next year's workshop. Check out the papers that are available on the USENIX Online Library:
<http://www.usenix.org/publications/library>.

Thanks to Gerald Lester, Mike Hopkirk, and Ken Corey whose workshop notes helped in the preparation of this report. More information about Tcl/Tk can be found at *<http://www.sunlabs.com/research/tcl/>*.
Next year's workshop will be held in July and a Call for Papers will be out shortly.

The Demise of *Computing Systems*

by Andrew Hume, President, USENIX Association
<andrew@usenix.org>

Computing Systems is the journal published by USENIX. Since its inception in 1988, it has been dedicated to the theory, design, art, engineering, and implementation of advanced computing systems. At the time of its conception, the USENIX Board of Directors felt a need for a journal that would publish papers of a practical rather than theoretical nature and be of value five or more years into the future. At that time there were only a couple of publications that met these criteria or that published much within the area of UNIX and UNIX-like systems.

Happily, there are now many more venues for such papers. Unhappily, the paper submission rate for *Computing Systems* has dropped off markedly over the last few years. In January, the USENIX Board of Directors decided to try to raise the submission rate by directly soliciting authors, including a letter from the editor, Dave Presotto, which was sent to the members of USENIX. The results were such that Presotto has recommended that we cease publication of *Computing Systems* after the final issue of the current volume (9:4). The board has accepted his recommendation.

Computing Systems has produced many outstanding issues, and we owe much to the vision, energy, and dedication of both Mike O'Dell, the editor in chief for the first seven volumes, and Peter Salus, who has served as managing editor since its inception. We also extend our thanks for the often overlooked contributions of the Editorial Advisory Board and the recent efforts of Dave Presotto as editor.

In Search of a Better Internet Language

by John E. Schimmel
<jes@sgi.com>

With all the hype around the Internet in the last year, there are a number of projects that have received a strong shot in the arm. Perhaps the best known of these now is Java, which was a language Sun developed for the embedded market and interactive television. It gained new life as an Internet programming language. Another of these is Tcl, another interpreted language useful for Internet programming. Lately, Tcl received its chance when Sun released a modified Tcl interpreter as a Netscape plug-in library.

Tcl was written by John Ousterhout while he was at the University of California, Berkeley, and achieved great fame as a simple, easily embeddable, interpreted programming language. When the Tk X windows bindings were added to the language it became a very simple way to create X windows applications which were portable to many different UNIX machines.

Since moving to Sun, Ousterhout has put together a team that has abstracted the graphical programming aspects of Tcl/Tk and ported the language to Apple Macintosh and Microsoft Windows environments. And now they have

embedded the language into a Netscape plug-in library so that scripts can be placed in HTML pages on WWW servers and executed in a browser window on nearly any UNIX, Windows, or Macintosh system.

Netscape and other browsers have the ability to load in extensions as shared libraries and give control of part of the browser window to the library code. This enables the browser to support a large number of data types mixed in the same window without the weight of embedding the support of all possible data types directly into the browser executable. The plug-in feature was first created for the Java interpreter so that Java code could run in a window; it was later generalized to support any data type.

There are now a number of different languages that you can embed directly into a Web page, including Java and Tcl. Each of these has strengths and weaknesses, so developers still need to know what is the right tool for their particular jobs. The primary strengths of Tcl as an Internet programming language are that it has a well-developed API to many windowing interfaces, it is very concise and simple to use, it already runs on nearly every popular platform, it can be used as a shell for interactive development and use, and it has the weight of several hundred thousand programmers developing applications in it for the past several years.

Unfortunately, Sun has not released the plug-in in source code form, so locally extending the language with C code is impossible. Perhaps this policy will change in the future.

I strongly encourage everyone to compare for themselves the use of Tcl/Tk as an Internet language. The plug-in and a large collection of example code is freely available from: *<http://www.sunlabs.com/research/tcl/plugin>*.

IPv6 & IPsec Software Available

by Ran Atkinson
<rja@inet.org>

The July 1996 NRL IPv6 and IPsec software release for BSD is now available online from *<http://web.mit.edu/network/isakmp>* in the US, and an exportable version is available from *<ftp://ftp.ripe.net/ipv6/nrl/>* in Europe. This software package was described in a paper in the January 1996 USENIX Conference and will drop into 4.4 BSD, 4.4-Lite BSD, NetBSD current, and BSDI 2.1. It features IPv6 support, IP security for IPv6, IP security for IPv4, the PF_KEY key management API, and the supporting Key Engine for the kernel. The exportable version omits the encryption software but does still include cryptographic authentication via the IP Authentication Header. Sites wanting to get early experience with IPv6 might find the NRL software package a good method for doing that.

Cisco Systems has created a freely distributable implementation of the "Internet Security Association & Key Management Protocol (ISAKMP)" with the "Oakley Key Exchange (Oakley)" for BSD UNIX. This implementation uses the PF_KEY API from the NRL software distribution to communicate with the kernel. This ikmpd(8) software from cisco is a daemon that opens a PF_KEY socket to communicate with the kernel and uses PF_INET sockets to communicate over UDP/IP to remote nodes. ISAKMP and Oakley are being considered by the IP Security Working Group of the IETF for possible future publication as standards-track RFCs. Cisco's implementation is designed to provide a proof of concept that ISAKMP + Oakley can work well for the Internet and is also intended to help get Internet security deployed more quickly by making dynamic key management available. This cisco software is available from an easy to use Web form at either *<http://web.mit.edu/network/isakmp/>* or from *<http://www.cisco.com/public/library/isakmp/isakmp.html>*. Export restrictions make it currently unavailable outside the US and Canada.

CMU Andrew Consortium Releases C++ Version Of Andrew UI and WP System

by Anne Watzman
<aw16+@andrew.cmu.edu>

Carnegie Mellon University's Andrew Consortium has released the first C++ version of its Andrew User Interface System (Andrew7.4), an integrated suite of compound document applications with which UNIX users can create documents containing combinations of text, pictures, graphs, figures, spreadsheets, and other embedded objects.

The new software, which is available free of charge, has been released in binary form to simplify its installation. The entire suite of Andrew7.4 applications can be found on the World Wide Web at *<http://www.cs.cmu.edu/~AUIS>*.

Andrew is more than a word processor. It includes a drawing editor, image displayer, mail/bulletin board reader/writer/manager, documentation browser, and numerous other capabilities. It is extensible in C++ and its own scripting language. The Andrew environment has ability to seamlessly

cut, copy, and paste compound text and embedded objects from one document into another.

The Andrew Consortium helps users to take full advantage of the system, including training, documentation and distribution options other than the web. A comprehensive user's guide is available for $25.
For additional information, contact AnnMarie Zanger,
phone 412 268 6710,
email <*zanger@cs.cmu.edu*>
or contact
Andrew Consortium
School of Computer Science
Carnegie Mellon University,
5000 Forbes Avenue,
Pittsburgh, PA 15213.

International Olympiad in Informatics Results

Competing against 215 students from 57 countries, three high school students representing the United States won medals in the 1996 International Olympiad in Informatics (IOI), held in Veszprem, Hungary, July 25-August 1.

Daniel Adkins, 16, from McKinley High School, Baton Rouge, LA, ranked 46th in the contest and won a silver medal. Matt Craighead, 14, St. Paul Academy and Summit, St. Paul, MN ranked 71st and received a bronze medal. Keldon Jones, 18, Oklahoma School of Science and Math, Oklahoma City, OK, ranked 96th and received a bronze medal. Joseph Turian, 16, Great Neck High School, Great Neck, NY, also competed for the US.

In the medal rankings, the USA team came in 22nd. The top five teams were: China, Russia, Slovakia, Poland, and Romania. Don Piele, professor of mathematics at the University of Wisconsin-Parkside, and leader of the United States team, reported that the Eastern European countries are extremely strong in algorithmic computer problem solving because of the many competitions held in this region. "They are consistently in the top-ranked teams at IOI," Piele noted.

During the IOI competition, students compete on two days. They are given five hours to use logic, mathematics, and computer programming skills to create original computer programs that solve three difficult problems.

Since 1992, when the US first entered the IOI, participating US teams have won four gold medals, five silver medals, and eight bronze medals. "We slipped a bit this year," Piele conceded, "when Joseph (Turian) failed to get a medal. But, the students enjoyed another exciting IOI."

USENIX is proud to be the sponsor of the United States of America Computing Olympiad (USACO). For more information about this exciting program, we recommend you visit the USACO Web site: <*http://usaco.uwp.edu/*>.

USACO Finalists at IOI, Veszprem, Hungary. Left to right: Keldon Jones, Daniel Adkins, Matt Craighead, Joseph Turian.

SAGE, the System Administrators Guild, is dedicated to the advancement and recognition of system administration as a profession. SAGE brings together system and network administrators for:

- professional and technical development,
- sharing of problems and solutions,
- communicating with users, management, and vendors on system administration topics.

SAGE News Editor

- Tina Darmohray
  <tmd@usenix.org>

SAGE Board of Directors

- Paul Evans, President
  <ple@usenix.org>
- Tim Gassaway, Secretary
  <gassaway@usenix.org>
- Barb Dijker, Treasurer
  <barb@usenix.org>
- Helen Harrison
  <helen@usenix.org>
- Bryan McDonald
  <bigmac@usenix.org>
- Hal Miller
  <halm@usenix.org>
- Kim Trudel
  <kim@usenix.org>

SAGE Working Groups

Group	Chair
sage-certify	Paul Moriarty
sage-edu	Ron Hall
sage-ethics	Hal Miller
sage-jobs	Tina Darmohray
sage-locals	Rene Gobeyn
sage-online	Pat Wilson
sage-policies	Lee Damon

You can contact these groups via email at
<group-name@usenix.org> for example, *<sage-certify@usenix.org>*.

SAGE Discussion Groups

sage-security
sage-managers
sage-outreach
sage-pt
sage-solo

SAGE BOFs

sage.board
sage.bof.women

SAGE Online Services

Email server:
majordomo@usenix.org

FTP server:
<ftp.sage.usenix.org >

WWW URL:
<http://www.sage.usenix.org>

SAGE Supporting Members

Bluestone, Inc.
Enterprise Systems Management Corp.
Great Circle Associates
Pencom Systems Inc.
Southwestern Bell
Taos Mountain

SAGE NEWS

Not Fancy, Just Functional

by Tina M. Darmohray
<tmd@iwi.iwi.com>

On Saturday night my husband and I met two friends in San Francisco at one of our favorite restaurants. We pulled up and found a parking spot right in front. As we walked in, we were told they were without electricity (explains the "good luck" we had parking) due to the power outage that was affecting many of the western states (and some of San Francisco). It was disappointing.

We decided to regroup. We knew that not all of S.F. was without power, so we decided to find another restaurant close by. We asked for a telephone book. They warned us not to use their pay phone because it had taken their money but not placed their calls. (We hypothesized how the pay phone might need electricity to process the coins properly.)

Luckily, one of our friends had brought his toolbag with him. A diligent sysadmin, he had a cell phone, some tools, and a tiny flashlight. Nothing really fancy, but, as it turned out, pretty functional. We used the cell phone to verify that our next choice of restaurant was up and running and not inundated. Next, some of us took turns using the flashlight and the facilities. With the aid of the toolbag, we were prepared to head out again.

Over dinner we discussed the power outage. How can such a "production" infrastructure be wiped out by a tree root or sagging wires? There must be a better way to engineer those circuits so they wouldn't be so interdependent. And this led into comments on the recent AOL debacle. It raises the same sort of issue. You'd think that such online organizations would plan, test, and design backup strategies, and redundancies, ad nauseam. We made the predictable comments that they all needed some good system administrators.

We returned home to find that, for this round of power outages, our home was spared. Nevertheless, we changed the oil in our generator and returned it to our designated "emergency preparedness" shed, where it awaits the winter "storms" when it will be pressed into service.

Sometimes management and higher-ups get carried away with form over function. It can be hard to convince them to let you budget for the time that is *really* needed to do a cautious and organized network upgrade, for instance. All too often, the tendency is to justify the short-term chaos that is created by the ends we're trying to achieve "right now." I tend to favor good, clean, even boringly-redundant, network design and planning to the immediate desire for the latest whiz-bang application. I guess that approach can have drawbacks too, but sometimes not-so-fancy baseline functionality is really great!

Jack Of All Trades

by Eric Berglund
<berglund@Synopsys.COM>

It occurs to me that sometimes network and system administration is like:

Detective Work

Cases get opened all the time. Resources are assigned in proportion to the seriousness of the situation, the likelihood of efficient diagnosis and repair (or solution and implementation), or the political visibility of the case.

If a solution is not forthcoming in a timely manner, the problem gets less and less attention, until finally it either sits in limbo with no resources or new evidence comes to light. When the latter happens, the case is reactivated and reevaluated for resource allocation.

My thought is that this characterization is not only descriptive, but should perhaps be approved and appreciated. Of course, it's frustrating for crime victims to have their case slowly relegated to the outlying file cabinets, but likelihood of success is a strong component of return on investment and thus an important factor in wise decisions about resource use.

Cafeteria Provision

None of us thinks twice about criticizing cafeteria food. It is a topic no more fraught with emotion or conflict than the weather–a pleasant conversational amusement.

In reality, there are people in facilities departments and people working for the vendor who take pride in the selection of menu and ingredients and in the ways they have "worked smarter" to deliver the best product possible within very restrictive resource constraints. Those people are invisible to us; they are expected to produce a service so seamless that it imposes no overhead or sense of obligation upon its beneficiaries.

It occurs to no one to investigate their choices rather than criticize them. Doing so would be considered a waste of time, even mildly distasteful. And if one does ask a question or wish information, the organizations are so streamlined that there's likely to be no one available for a time-consuming discussion of alternatives and trade-offs.

Nonetheless, despite their invisibility and seeming invulnerability ("After all, if they think that's food, they can't really be that sensitive, can they?"), the best will stay only where their efforts are appreciated, praised, supported, and rewarded. The true experts will find situations in which their clients are discriminating enough to recognize quality, make it a priority, pay for it, and communicate their gratitude when it is achieved.

Architecture

The architect's client has needs, desires, opinions, and whims, all of which must be considered and respected. Nonetheless, most clients are unaware of structural, mechanical, electrical, logistical, safety, and legal constraints. Most believe that the "important" parts of architecture (the aesthetic ones) are something they could do as well as the architect, if they didn't have other things to do. And, of course, most think the architect's fee is overly high and the cost of his materials inflated.

The most knowledgeable clients are capable of selecting an excellent architect, accepting the costs of quality, and accelerating the process by clearly specifying and prioritizing their needs and desires. They may even make suggestions that inspire or guide the architect.

Cooperative but less expert clients let the architect guide the decision-making process. They present the project, describing what they can about their constraints and preferences. The architect then presents reasonable and realistic alternatives, letting clients make the choices where the project permits them. The clients ask questions when they are concerned about the choices, and the architect willingly answers in terms the clients can understand. The process of communication is a cooperative two-way street, not a series of demanded and produced reports justifying every decision and anticipating every objection. Clients avoid steadfast attachment to their own preconceptions, knowing that as their knowledge increases, they may have to revise their thinking.

The difficult–and ultimately undesirable–clients are those who think they know more than they do, treat the relationship as adversarial, and seldom use questions to understand. Instead, their queries are accusations, and they excuse their behavior as an insistence on high standards. In the end, they are satisfied only when they feel they have controlled the situation and dominated the architect. These clients typically overconstrain the solution, leaving too few resources for solid infrastructure or missing opportunities for efficiency or elegance.

Ignorant and dismissive clients know little and want to know no more. They want the project done without their involvement or commitment and are annoyed or even angry when forced to consider constraints other than the ones of which they are already aware. Their information sources are second- or thirdhand and their attention is randomly focused. Their lack of participation with or trust in the architect

causes delays and distractions in the best of cases, recrimination and expensive rework in the worst.

In the end, clients bear primary responsibility for the quality of the work: they must choose the architect wisely, reach a clear understanding of their alternatives, communicate clearly their priorities and funding commitments, and work cooperatively with the architect to clarify and modify decisions as the project proceeds. (Clients are, of course, free to delegate the collaborative decisions fully to the architect, but they cannot delegate their responsibility for the end product.)

Watching Browser Technologies

by Shawn Instenes
<shawni@siforest.com>

I confess. I come from a sysadmin background that includes some pretty paranoid places. I worked at one site where they scan every bit of machine-readable media that comes in–floppies, CD-ROMS, tapes, what have you–for viruses. I assume that would include paper tape, but I haven't challenged them on this point. I imagine it would take a while to find a working reader.

Although most sites don't expend that much time and effort to maintain their data integrity, they *do* have some sort of policy (or should have): for example, it's common to restrict bringing in floppies from home.

Now I wonder about browsers. Only recently have browsers been extended to retrieve what I call "executable content": programs of one sort or another, rather than images and text. Plug-ins, Java, JavaScript, ActiveX, all are methods of extending browser functionality in one way or another. Does your site have a data integrity policy? Is it watching these technologies, too? They're just as much a vector of transmission as floppies are.

There are degrees of danger here. JavaScript is interpreted, and not translated into machine code. Java can be converted to machine code to improve its speed. It also can include "native" methods, which are subroutines written in machine code. Plug-ins are machine code, but they aren't installed automatically. . . yet. They've been written for some pretty interesting things (see the URLs list below for examples). ActiveX components are machine code, too, and they can be installed automatically. The browser is supposed to prompt the user before it does this, but nothing (in the specification) says it is required to do so.

To allay concern over running arbitrary code downloaded over a network, there are some efforts to verify the integrity of the code along with its origin. An extension signed by this standard can be proven not to have been modified since it was signed, and the author can be identified positively. You can be assured that this really did come from who is says it does.

But where's the security? Do vendors never write buggy code? This plan makes no assurances about the security of these programs at all.

Let's consider a hypothetical situation in which a company develops a browser extension that can process an already existing word processing format. This theoretical company reuses existing code (that's good programming practice, right?) so that this extension acts in many ways just like the product they've been selling. This word processing program includes a macro capability. Ooops. Now we have the possibility of "hostile macros," perhaps one written to edit `.rhosts` files or your resume.

Perhaps that's far-fetched. The point is that there is nothing to prevent buggy code (example: it uses fgets()) or just badly designed code (example: macros can be automatically executed without the user knowing) from being run on your browser, because a vendor's approved code release may be both of these! Both bugs and unwanted features may provide an avenue for an attacker to do mischief. It's something to watch.

URLs to Visit

The Exploder ActiveX control:
<http://www.halcyon.com/mclain/ActiveX/>

Microsoft's Authenticode technology:
<http://www.microsoft.com/intdev/signcode/>

Netscape's Plug-in design specifications:
<http://home.netscape.com/comprod/development_partners/ plugin_api/plugin_design.html>

Sun's Java security:
<http://www.javasoft.com/sfaq/>

SunLab's Tcl plug-in:
<http://www.sunlabs.com/research/tcl/plugin/>

Perl Practicum: Network Wiles (Part II)

by Hal Pomeranz
<hal@netmarket.com>

In the last installment, we saw how to program a network client by writing a simple tool to get pages from remote Web servers. In this issue, we will explore how to write a simple network server. As an example project, we will actually write a simpleminded Web server (the complete code is presented at the end of this article in case you find it easier to follow along that way). Reread the previous issue if you think you have forgotten any of the basic networking concepts I presented there.

Getting Started

The first thing a network server must do is set up a socket upon which it can accept requests. The first phase of this process looks a lot like the initial code of a network client:

```
use Socket;

$this_host = 'my-server.netmarket.com';
$port = 8080;

$server_addr =
        (gethostbyname($this_host))[4];
$server_struct =
        pack("S n a4 x8", AF_INET, $port,
        $server_addr);
$proto = (getprotobyname('tcp'))[2];
socket(SOCK, PF_INET, SOCK_STREAM, $proto)||
  die "Failed to initialize socket: $!\n";
```

First, the program has to pull in the Perl `Socket.pm` module. The hostname of the machine upon which the server will run and the port upon which it will accept requests are specified on the next two lines (you can imagine getting these parameters out of a configuration file or on the command line). The program then calls `gethostbyname()` to get the IP address of the server machine and uses that information to create a C structure which we will use later. Finally, we call `socket()` to create a file handle for the socket.

Remember from the last article that Web servers usually wait for connections on port 80. Why does the code above specify the port as 8080? As a security feature, only the superuser is allowed to run servers that accept connections on ports below 1024. The thinking behind this policy is that users should then be able to trust connecting to unknown machines as long as they are connecting to services (like Telnet, FTP, gopher, et al.) that listen for connections at low port numbers because they will require the system manager at the remote site to "approve" the service being run on those ports. This reasoning is probably no longer true in this age of workstations on every desk, but the rule remains.

Returning to our example, the server now needs to prepare to receive connections at the given address and port combination:

```
setsockopt(SOCK, SOL_SOCKET, SO_REUSEADDR,1)
        ||
  die "setsockopt() failed: $!\n";
  bind(SOCK, $server_struct) ||
        die "bind() failed: $!\n";
listen(SOCK, SOMAXCONN) ||
        die "listen() failed: $!\n";
```

The `setsockopt()` function allows the program to change various parameters associated with the socket: more on `SO_REUSEADDR` in a moment. The `bind()` call is what actually associates the `SOCK` file handle with the address and port number pair specified at the top of the program. As long as any program has bound itself to a particular address and port, no other program can bind to the same location. This is useful and prevents confusion. However, even after a given server program has exited, its address/port combination does not become available for reuse (at least until the machine the server was running on is rebooted)–even if you rerun the exact same program. This is annoying and creates bad feelings. Use `setsockopt()` to set the `SO_REUSEADDR` bit to 1 (true)–BEFORE the call to `bind()`–so other programs can reuse the same port after the server program has exited. Both the `SOL_SOCKET` and `SO_REUSEADDR` constants are defined in `Socket.pm`.

The `listen()` call is probably misnamed. All this function does is specify how long a queue of pending connection attempts the server is willing to deal with. If the server queue is full, further connection attempts will be rejected. On almost every socket implementation in existence, the maximum queue length that you can set is 5 (so handle incoming connection requests quickly!), and `SOMAXCONN` (another helpful constant from `Socket.pm`) is usually set to 5. If you try to set the queue length to a value above 5, the operating system silently throttles the queue length back to the maximum value. Solaris 2.x is the only modern operating system that I am aware of where you can meaningfully specify queue length values that are greater than 5 (though interestingly `SOMAXCONN` is still given as 5 in the Solaris 2.x system header files).

Dealing with Pending Requests

At this point, most network servers go into a tight loop so that they can rapidly deal with their queue of pending network connections:

```
for (;;) {
    $remote_host = accept(NEWSOCK, SOCK);
    die "accept() error: $!\n"
    unless ($remote_host);

    # do some work here

    close(NEWSOCK);
}
```

The `accept()` call grabs the next connection request off the pending queue for `SOCK`. (If there are no pending connections, `accept()` pauses until one comes in.) A new socket that is the local endpoint of this new communications channel is created. If you print to `NEWSOCK` you are sending data to the remote machine making the connection, and you can read data from `NEWSOCK` just like any other file handle to get data from the remote machine. Always remember to close `NEWSOCK` when it is no longer needed.

The `accept()` function returns a C structure containing the address of the remote machine (or `undef` if the `accept()` fails for any reason). This structure is the same as the one passed to `bind()` and `connect()`, and you can extract the IP address of the remote machine as follows:

```
$raw_addr =
    (unpack("S n a4 x8",$remote_host))[2];
@octets = unpack("C4", $raw_addr);
$address = join(".", @octets);
```

You can also obtain the hostname of the remote host (usually) with the `gethostbyaddr()` function:

```
$hostname =
    (gethostbyaddr($raw_addr,AF_INET))[0];
```

This can be useful for logging purposes. Note the reappearance of `AF_INET` – `gethostbyaddr()` needs to be told what type of network address it is being given.

A Simple Web Server

Up to this point, we've been flushing out the basic skeleton that every network server application has to have. Now let's do something interesting with it.

HTTP is an incredibly simpleminded protocol. Requests sent by the Web browser are simply lines of ASCII text, terminated by a blank line. After seeing the blank line, the server sends back the requested data and shuts down the connection. Although the client typically sends over a great deal of useful information in its request, a simple Web server can ignore everything except the line that looks like:

```
GET /some/path/to/file.html ...
```

Here's some code that reads the client request and extracts the path to the information that the user is requesting:

```
while (<NEWSOCK>) {
    last if (/^\s*$/);
    next unless (/^GET /);
    $path = (split(/\s+/))[1];
}
```

Now the server has to respond. Typically `$path` is relative to the top of some directory hierarchy where your Web documentation lives–your `$docroot` in Web-speak. This directory can be defined in a `config` file or on the command line. Assuming that `$docroot` has been defined elsewhere we can simply

```
if (open(FILE, "< $docroot$path")) {
    @lines = <FILE>;
    print NEWSOCK @lines;
    close(FILE);
}
else {
    print NEWSOCK <<"EOErrMsg";
<TITLE>Error</TITLE><H1>Error</H1>
The following error occurred while
trying to retrieve your information:
$!
EOErrMsg
}
```

If we are able to open the requested file, we simply dump its contents down `NEWSOCK`. Note that the server sends back an error message if the `open()` fails. Never forget that there is somebody on the other end of that connection who is waiting to hear something back as a result of his or her request.

Congratulations. If you glue together all the code fragments in this article, you will have a bare-bones Web server. You will find all of the code in proper order at the end of this article to make it easier to review all the concepts presented here.

That's Not All

Although this Web server "works" as far as answering simple requests for information, it has a number of problems. First and foremost, it only can handle one request at a time: most production-quality servers can handle hundreds or thousands of simultaneous requests. Second, if you run this server on your machine, I can request

```
/../../../../../../../etc/passwd
```

and get a copy of your password file. Obviously, a better access control mechanism is needed.

In the third and final installment of this series, we will look at ways to solve these (and other) problems with our mini Web server.

```
#!/packages/misc/bin/perl

use Socket;

$docroot = '/home/hal/public_html';
$this_host = 'my-server.netmarket.com';
$port = 8080;

# Initialize C structure
$server_addr =(gethostbyname($this_host))[4];
$server_struct = pack("S n a4 x8", AF_INET,$port, $server_addr);

# Set up socket
$proto = (getprotobyname('tcp'))[2];
socket(SOCK, PF_INET, SOCK_STREAM,$proto)||
    die "Failed to initialize
        socket:$!\n";

# Bind to address/port and set up pending queue
setsockopt(SOCK, SOL_SOCKET, SO_REUSEADDR, 1) ||
    die "setsockopt()
        failed: $!\n";
bind(SOCK, $server_struct) || die "bind() failed: $!\n";
listen(SOCK, SOMAXCONN) || die "listen() failed: $!\n";

# Deal with requests
for (;;) {
    # Grab next pending request
    #
    $remote_host = accept(NEWSOCK, SOCK);
    die "accept() error: $!\n" unless ($remote_host);

    # Read client request and get $path
    while (<NEWSOCK>) {
        last if (/^\s*$/);
        next unless (/^GET /);
        $path = (split(/\s+/))[1];
    }

    # Print a line of logging info to STDOUT
    $raw_addr = (unpack("S n a4 x8", $remote_host))[2];
    $dot_addr = join(".", unpack("C4", $raw_addr));
    $name = (gethostbyaddr($raw_addr, AF_INET))[0];
    print "$dot_addr\t$name\t$path\n";

    # Respond with info or error message
    if (open(FILE, "< $docroot$path")) {
        @lines = <FILE>;
        print NEWSOCK @lines;
        close(FILE);
    }
    else {
        print NEWSOCK <<"EOErrMsg";
<TITLE>Error</TITLE><H1>Error</H1>
The following error occurred while trying to retrieve your information: $!
EOErrMsg
    }

    # All done
    close(NEWSOCK);
}
```

New SAGE Booklet Now Available

The second booklet in the SAGE "Short Topics in System Administration" series, *A Guide to Developing Computing Policy Documents* has been mailed to all SAGE members and is also available for purchase.

Edited by Barbara L. Dijker <*barb@usenix.org*>, the *Guide* addresses key issues of site management: why a site needs policies, what a policies document should contain, who should draft it, and to whom it should apply. Topics covered include:

- Writing Guidelines
- Developing Policy Documents
- A Detailed Document Outline
- Security
- Safety
- Facilities and Resources
- Use Agreement

Sample documents, a reference guide and related resources are also included.

Developed with help from the SAGE Policies Working Group, Dan Appelman, Lee Damon, Rob Kolstad, Hal Miller, Wes Morgan, David Parter, Mary Seabrooke, Elizabeth Zwicky, and the USENIX staff the goal of the "Policies Booklet" is, as Barb writes in her introduction, "to provide a comprehensive guide to developing computer use policies that everyone within the organization will be pleased to endorse."

Additional copies may be obtained from the USENIX office <*office@usenix.org*>. The price is $5.00 each (for SAGE members) or $7.50 each (for non members). Residents outside Canada and the US should add $3.50 per copy for postage.

For further information, please refer to the SAGE webpage <*http://www. usenix.org/sage*>.

BAY LISA Meeting News

The BayLISA (California) group meets monthly to discuss topics of interest to systems and network administrators. The meetings are free and open to the public.

BayLISA holds monthly meetings on the third Thursday of each month at 7:30 PM PST. The meetings are also broadcast via MBONE.

Note: AS OF OCTOBER, WE ARE MOVING TO CISCO, near Highway 237 and 1st Street. Stay tuned or check out the Web page for more info. Thanks very much to Synopsys for hosting us for all this time.

Schedule

October 17
Windows NT for the Unix Systems Administrator
N.K. Krishnan, Hewlett Packard Company

November 21
BayLISA Board Elections & member meeting before the regular meeting Talk to a current board member or send mail to <*blw@baylisa.org*> if you are interested in running.

December 19
Our special holiday meeting– suggestions being accepted.

For further information on BayLISA, check out our web site: *http://www.baylisa.org/*
To get further information on the meeting location, you can FTP it from <*ftp.baylisa.org:/BayLISA/location*> or query the BayLISA mail server by cutting and pasting the following line to your shell

```
echo "index baylisa" |
        mail majordomo@baylisa.org
```

BayLISA makes video tapes of the meetings available to members. For more information on available videos, please send email to: <*video@baylisa.org*>.

For any other information, please send email to: <*info@baylisa.org*>.

Defamation and the Net

by Nicholas S. Nassif
<nicholas@online2000.com>

Editor's Note: Nicholas S. Nassif will be writing occasional columns about legal issues and how they affect members of our community.

Your dream has finally come true. You have persuaded investors to finance you, and you have started Speechcom, a full-service Internet Service Provider, which includes a bulletin board service. You have worked 16 hours a day for three straight years and have built Speechcom's customer base to 20,000. The break-even point is 16,000 customers, so Speechcom is showing a modest profit.

Your day begins typically; then a sheriff comes into your office and serves you with a lawsuit. You immediately call your attorney to investigate the allegations by the plaintiff, Melrose Ltd.

You discover that the Melrose Ltd. is a securities investment banking firm. Speechcom runs a nationwide, commercial online service aimed at the general public at large. A member of the Speechcom network anonymously posted a message to the Speechcom Money Views bulletin board claiming that Melrose Ltd. engaged in criminal and fraudulent conduct in connection with a public stock offering. Melrose Ltd. is claiming that Speechcom is a "publisher" of the Money Views bulletin board and is therefore liable for its content.

You then sit with your attorney to determine the possible liability of Speechcom. Your attorney asks you if Speechcom has established any editorial guidelines for Money Views bulletin board to follow. You ask your attorney why that is relevant. Your attorney asks if you have a few minutes. You answer, "Of course." Your attorney then enlightens you as to the state of defamation law in cyberspace.

The first major case in the area of defamation disputes was a 1991 case entitled *Cubby v. CompuServe* ("CompuServe case"). CompuServe developed, among other services, CompuServe Information Service ("CIS"), an online general information service or "electronic library" that paying subscribers could access from a personal computer. Subscribers had access to literally thousands of information sources, available on CIS. Subscribers could also obtain access to more than 150 special interest "forums," including electronic bulletin boards, topical databases, and interactive online conferences.

One forum available on CIS, the Journalism Forum, focused on the journalism industry. Cameron Communications Inc. ("CCI"), which was independent of CompuServe, contracted to "manage, review, create, delete, edit, and otherwise control the contents" of the Journalism Forum "in accordance with editorial and technical standards and conventions of style as established by CompuServe."

Rumorville USA ("Rumorville") was a daily newsletter that provided reports about broadcast journalism and journalists and was a publication that was made available on the Journalism Forum. CompuServe had no employment, contractual, or other direct relationship with the publisher of Rumorville. The contract between CCI and the publisher of Rumorville ("Publisher") provided that the publisher "accepted total responsibility for the contents" of Rumorville.

CompuServe had no opportunity to review Rumorville's contents before the Publisher uploaded it into CompuServe's computer banks, from which it was immediately available to approved CIS subscribers.

CompuServe received no part of any fees that the Publisher charged for access to Rumorville. The compensation CompuServe received for making Rumorville

available to its subscribers was the standard online time usage and membership fees charged to all CIS subscribers. CompuServe had no notice of any complaints about the contents of Rumorville.

In 1990, plaintiffs Cubby Inc. ("Cubby") and Robert Blanchard ("Blanchard") (collectively, "plaintiffs") developed Skuttlebut, a computer database designed to compete with Rumorville. Subscribers gained access to Skuttlebut through their personal computers after completing subscription agreements with plaintiffs.

Plaintiffs claimed that, on separate occasions in April 1990, Rumorville published false and defamatory statements relating to Skuttlebut and Blanchard and that CompuServe carried these statements on its Journalism Forum. Plaintiffs sued CompuServe and other defendants, among other causes of action, for defamatory statements contained in Rumorville.

CompuServe argued that, "based on the undisputed facts, it was a distributor of Rumorville, as opposed to a publisher of the Rumorville statements." CompuServe further contended that, as a distributor of Rumorville, it could not be held liable on the libel claim because it "neither knew nor had reason to know of the allegedly defamatory statements."

The underlying laws in New York were deep rooted. According to *Cianci v. New Times Publishing Co.*, a 1980 case ("Ciani case"), "one who repeats or otherwise republishes defamatory matter is subject to liability as if he had originally published it."

Following the Ciani case was *Lerman v. Chuckleberry Publishing, Inc.*, a 1981 case. The court held that with respect to news vendors, book stores, and libraries, however, "New York courts have long held that vendors and distributors of defamatory publications are not liable if they neither know nor have reason to know of the defamation."

In the CompuServe case, the court reasoned that CompuServe's CIS product was "in essence an electronic, for-profit library that carried a vast number of publications and collected usage and membership fees from its subscribers in return for access to the publications." Once CompuServe decided to carry a publication, CompuServe had little or no editorial control over that publication's content. That was especially true when CompuServe carried a publication as part of a forum that was managed by a company unrelated to CompuServe.

The court continued and ruled that CompuServe had no more editorial control over Rumorville than "does a public library, book store, or newsstand, and it would be no more feasible for CompuServe to examine every publication it carries for potentially defamatory statements than it would be for any other distributor to do so."

The court continued, "A computerized database is the functional equivalent of a more traditional news vendor, and the inconsistent application of a lower standard of liability to an electronic news distributor such as CompuServe than that which is applied to a public library, a book store, or a newsstand would impose an undue burden on the free flow of information. Given the relevant First Amendment considerations, the appropriate standard of liability to be applied to CompuServe is whether it knew or had reason to know of the allegedly defamatory Rumorville statements."

CompuServe contended that it had neither knowledge nor reason to know of the allegedly defamatory Rumorville statements, especially given the vast number of publications CompuServe carries and the speed with which Rumorville is uploaded. The plaintiffs brought forth no evidence to the contrary.

The court agreed with CompuServe and granted CompuServe's summary judgement motion. It held that CompuServe was not held liable because CompuServe was a "distributor" and not a "publisher." The court concluded that because CompuServe did not actively monitor the postings of the forum, it was a distributor. CompuServe would not be held to the "strict standard" of a publisher, but instead it would be held to a lower standard and would be liable only if CompuServe "knew or should have known" that the defamatory article was posted. In summary, the court compared CompuServe to a bookstore selling the book rather than the publisher of the book.

After hearing the court's holding in the CompuServe case, you are relieved. However, your attorney has not completed his lecture to you on the state of the defamation law. The *Stratton Oakmont v. Prodigy* case is the next landmark case that your attorney explains.

In 1995, in the case of *Stratton Oakmont v. Prodigy* ("Prodigy case"), the Supreme Court for Nassau County, New York, sent shock waves throughout the online world by denying a summary judgment motion by Prodigy, which relied on the *Cubby v. CompuServe* case and felt very confident it would prevail. Prodigy was sued because of alleged defamatory statements made by a customer of Prodigy on Prodigy's Money Talk service.

The court analyzed the facts and stated that the plaintiffs first had to establish that Prodigy was a publisher because one who repeats or otherwise republishes a libel is subject to liability as if he or she had originally published it. In contrast, distributors, such as bookstores and libraries, are liable for defamatory statements of others only if they knew or had reason to know of the defamation, which was the holding in

the CompuServe case. A distributor, or deliverer of defamatory material, is considered a "passive conduit" and is found liable in the absence of fault. However, a newspaper or magazine is more than a "passive receptacle or conduit for news, comment and advertising."

The critical issue in Prodigy was whether Prodigy exercised sufficient editorial control over its computer bulletin boards to render it a publisher with the same responsibilities as a newspaper or magazine.

The court in the Prodigy case reasoned that there were two distinctions between CompuServe and Prodigy. First, Prodigy held itself out to the public and its members as controlling the content of its computer bulletin boards. Second, Prodigy implemented this control through its automatic software screening program and established guidelines that board leaders were required to enforce. Prodigy was clearly making decisions as to content, and such decisions constituted editorial control.

Prodigy was held to be a publisher rather than a distributor. Prodigy's conscious choice, to gain the benefits of editorial control, has opened it up to a greater liability than CompuServe and other computer networks that make no such choice. The court, in denying Prodigy's summary judgment motion, reasoned that Prodigy held itself out as a "family" computer service and that it cautiously exercised editorial controls so that the content would be acceptable to families.

You then ask for a referral to a good bankruptcy lawyer. Your attorney says, "Not so fast. What type of editorial controls has Speechcom developed? The facts of which of the two cases fits Speechcom's situation?" Your attorney then goes through the facts of Speechcom's situation to determine potential liability.

It is important to discuss the editorial policy of your company and change those policies, if necessary, to limit your company's potential liability for statements made on your company's bulletin board services.

Please contact me with any questions.

Are Standards Worth the Effort?

by Nick Stoughton
<nick@usenix.org>

This article is the text of the opening argument given in the panel discussion at the September LISA conference on the question of "Are Standards Worth The Effort?" Future editorials will present the counter arguments.

To answer the question, we need to look at the effort that is put into building formal standards, and the benefits that they return. It is always a subjective question . . . is it worth the effort? And the answer will always depend on your personal viewpoint. Something that is highly beneficial to you, and does not cost you much, is clearly worthwhile. Something that has marginal benefit, and costs you considerable effort and/or money is of questionable worth. I want to spend some time here looking at what the effort is in producing a formal standard, and what are the benefits. You can then decide for yourself if it is worth *your* effort.

To understand and measure the benefits, let's consider some history. Ten years ago, the POSIX standardization effort had but started, and the first trial use system interface standard was in the process of being published. Few people ever used that standard as such, but it was the beginning of a major effort. Its successors, the 1988 and 1990 full use standards, the latter of which is an ISO international standard, have proved to be some of the most widely adopted information technology standards ever published. I don't know how many of you remember the pre-POSIX world, the System V versus Berkeley wars. The ability to buy a UNIX system from a dozen vendors, but have to port your code for each and every one. Every other line of your source was a "#ifdef" for a different architecture.

Of course, it meant big bucks for the software houses who ported applications from one place to another. But it was bad news for the end user, who had to pay for it.

The POSIX.1 and POSIX.2 standards did a massive amount to level that playing field. True, it has been said at USENIX conferences that there's no such thing as a portable application, even with POSIX, merely an application that's been ported. The testing effort to prove how an application behaves on a different platform is still omni-present, and still scares many software vendors from offering their application on more than a small handful of platforms. But the difference between systems has now come down to specific specialist areas that allow the hardware vendors to have their USPs.

The strength of POSIX is also one of its major weaknesses; it is entirely a volunteer effort. The POSIX standards are as

good as the people who write them make them. They are not employed to write standards, or at least, not by a standards development organization. Volunteers come together to write standards for areas that are perceived to be needing them. The people who perceive the weakness are the people who write the standard. There is no body sitting around saying "Hmmm . . .I think we need a standard for Print Administration in POSIX, let's see who will do one based on Palladium." Instead, a group of people come to POSIX and say "We want to write a Print Administration standard; it will be based on Palladium Will you let us call it POSIX when we are done?"

To get sponsorship within the POSIX process, that group has to follow the rules; they must be able to demonstrate the need for that standard (well, they did that by showing up); they must be able to show existing practice (POSIX is not a standardization by invention body), they must be able to show commitment to the process; and, they must follow the rules. Those rules include a ballot of the resulting document by any interested party (although in POSIX, strictly only members of the IEEE CS have binding votes), and they must achieve a 75% approval of the returned ballots (which in turn must be at least 75% of the ballot group).

So if standards sometimes seem to be at odds with what you expect, then it is because *you* did nothing to stop it being so. Anybody can be a member of a POSIX working group. Anybody can join the ballot group. OK, if we get 500 people in the ballot group, we might only concentrate on the C members, but that doesn't happen often, and it isn't really very expensive to become a member of the premier professional body (besides SAGE) for your profession!

This openness of process is POSIX's major strength. Currently, all POSIX standards are produced by individuals, not corporations. This has its failings, as we shall consider, but in general, is a Good Thing. I always thought of POSIX as being produced by a group of very special people, and I was unworthy to be part of it. But that simply isn't so.

Formal standards do have a lot of effort go into them. Sometimes, it feels like not enough; some standards take forever to come out. That is mainly through the lack of investment by the participants. The cost of membership is perhaps too low companies that spend $1,000,000 joining X/Open (each year) feel they need to put a lot of effort in making sure their investment is worthwhile. POSIX costs only perhaps $10,000, plus 5-6 weeks worth of time per year. This is so insignificant that many organizations begrudge the time to their employees, and the only place and time work happens is at the meetings.

What has to happen to make a formal, POSIX standard? As I said, the first step is for a group of people to decide that there's a need for a standard, some starting point or points, and a desire to do something about it. This group, often augmented by others once the project is sponsored, writes a document. That document is then sent out for formal ballot to the interested parties. This is the point at which real work actually starts. Many think that the step of writing the first draft is the hard part . . . but in reality that's the easy bit! Once you have a first draft, and several hundred or thousand comments, ranging from "please add a comma at the end of line 42" to "this standard is ill-conceived and based on entirely the wrong technology; you should start again," then the hard work starts. Get 75% of these people to say "Good job . . . I have no further objection."

So much for the cost of the effort. How many of you have reaped some benefit from formal standards? Well, how many of you have worked on systems from more than one supplier –say an HP and a Sun, or Silicon Graphics and IBM.All of you have benefited from POSIX. Didn't realize it? Well, RTFM some day! *Your* jobs are more secure because of POSIX. *Your* skills are more portable, as well as your applications. With no standards, or just proprietary ones, your skills would be tied to those systems you had direct experience of.

But perhaps the POSIX process is so imperfect that we should abandon it forthwith and start anew. How else could we organize things so that the results are more useful, more timely, more worthwhile? Actually, these questions have been taxing the governing body of POSIX this year. Is there something we can do to be more relevant? Are we missing the boat in some way? That particular debate is focusing on two areas: should the scope of POSIX be widened, and, should the process be altered in some way. The scope issue was initially thought to be the more relevant, but exactly no one showed up to talk about it at the meeting convened for the purpose last POSIX meeting. The process issue did attract many more. And yes, there are things we are considering very actively for changing the process; probably the biggest of which is corporate rather than individual membership for some projects. You feel kind of mean saying the reason that this standard is late is because Joe Soap was slow getting his actions done. But being able to say this standard is late because Sun Microsystems (naming no names) didn't fulfil their actions is a lot less personal (unless you work for Sun).

I have been fortunate enough to experience the way some other groups develop standards. The one I have been most active in is POSIX, but do other groups have a better way of working? Many ISO standards are developed by groups of people who represent not themselves, as in POSIX, or their company, but their country. All sorts of problems are introduced by this way of thinking. Huge compromises have to be made because people don't really want to create international incidents over something minor in a standard.

X/Open produces UNIX specifications, but charges the groups who participate (all companies or organizations) a

large fee. Small players and individuals cannot participate. Could *you* afford $25,000 to join one working group? Plus the cost of travel, your time, and other incidental costs? No, but your employer might. Of course, if you want to have real influence on the specification, you have to pay the full fee, a cool $1,000,000. That thins the crowd a bit.

The IETF have a low cost of membership, and do produce lots of informal standards as output. But that's the trouble. Their standards, Requests for Comment, are informal to a degree that is pretty nearly unenforceable. Is that really useful? Well, many could argue that the Internet is built on these informal standards and yes, it is pretty useful. It does seem that the IETF has something; some sort of sex appeal that makes people want to play their game. There is no denying that right now, the Internet *is* sexy. People will do things not to get a tick in the box for federal government approval (as is the case for POSIX), but to build something that people are going to use. Which is more sexy: a very informal procedure about what you should do if you think someone is attacking your site through the Internet, or, a formal standard about how printers should be administered? Clearly the first. Which is actually more useful to you? Now that's a bit harder. They are both useful to almost everyone in this room, provided they don't mean everything you've done and set up, outside those standards, over the past few years, should be immediately abandoned. How many people have contributed to the Incident reporting standard? Perhaps 50-60. How many to the print administration? Well, because the process is different, it is harder to measure like for like. But one measurement (the working group plus the ballot group) actually has the same sort of number. The IETF produce lots of small, informal standards each year. POSIX produces very few, big thick formal standards each year. Page for page, POSIX is only slightly behind the IETF. But because they are concentrated on a few areas, it looks like POSIX is producing substantially less than the IETF.

Both the IETF and POSIX are Open processes; anyone can help form that standard. X/Open, and to a lesser extent ISO, are essentially closed processes; you have to pay or be invited to join. If you're not a member, the only way you can object is to not adopt their standard.

To sum up, POSIX may not be perfect, but it does serve a useful purpose, and is in my opinion, worth 10 times the effort that is put into it.

Is Your Domain Name System Secure?

by John Gilmore
<gnu@toad.com>

The Domain Name System provides translations between domain names like *www.usenix.org* and numeric Internet addresses such as 131.106.3.1. It has become a largely invisible part of the Internet infrastructure, something that we all take for granted. We click on a link and the page comes up; it's so fast and reliable that we usually don't notice that DNS as well as HTTP was involved.

But behind the scenes, DNS is a high-performance, globally distributed, locally published, redundant, high reliability database system. It manages a huge hierarchical set of "resource records" of many types, providing fast access to all of them from anywhere on the net.

But the Domain Name System was designed when security was not an issue on the Internet. To be reliable in tomorrow's Internet, it has been retrofitted with cryptographic integrity checking.

Secure DNS provides protection against the spoofing of DNS records by parties other than those to whom the name space was delegated. It also provides a convenient infrastructure for the publication of keys or certificates, by any entity which desires to do so, for use in other protocols. For example, X.509 certificates, PGP keys, and keys used by IP-Security encrypting firewalls can all be published and accessed this way.

The DNS Security architecture involves off-line generation of keys and off-line signing of DNS resource records. These resource records are then moved to an on-net system for distribution. The signature process takes a standard DNS ASCII zone file as input, and produces a standard DNS ascii zone file as output, containing the original zone plus new SIG and NXT records. In low-security or prototype operations, key generation and record signing can be done on a machine on the Internet, accepting the risk that the private key will be compromised.

Full details on Secure DNS are available in the latest "dnssec-secext" Internet-Draft (at this moment, draft-ietf-dnssec-secext-10.txt), available wherever fine Internet-Drafts are sold (*ftp://ds.internic.net/internet-drafts/*).

BIND, the Berkeley Internet Name Daemon, is the main implementation of the Domain Name System protocols. Its distribution includes `named` (the name daemon), `libresolv` (the name resolution library), and various small tools such as `nslookup` and `dig`. Paul Vixie, of the Internet Software Consortium, maintains BIND.

Paul recently made the first public test release of BIND that contains partial DNS Security support, bind-4.9.5-T3B.tar.gz. (Current production and beta-test BIND releases are always available via <*http://www.vix.com/isc/bind.html*>.) This first release does not contain any cryptography, and I believe it to be completely unaffected by export controls. It adds the ability to handle KEY and SIG records in parsing, printing, queries, and zone transfers. It does not cryptographically validate these records. It also does not have code for the generation of KEY or SIG records. These facilities will be added in future releases. Any future release you find in this location will also permit worldwide export.

As a first step in DNS Security deployment, each site that wishes to experiment will need to run this version of BIND (or any later one). After one of your zones (e.g., your main domain) contains KEY or SIG records, all secondary servers for that zone will also need to run this BIND release or later. If they don't, zone transfers to those secondaries will fail (with a cryptic error message to `syslog`, probably ending up in `/var/adm/messages`). More detailed error messages can be found by running the name daemon (named) in debug mode; they will appear in files under `/usr/tmp/xfer.ddt.*`. But the error will be "unknown type" of DNS record, and the cure will be to run a later BIND, or remove the KEY and SIG records from the zone at the primary site.

The next step in deployment will involve using cryptographic code to generate experimental keys and signatures for some of your zones. KEY and SIG records can be created using Trusted Information Systems' prototype cryptographic Secure DNS implementation. It is only available to US citizens as of September 1, but check the file at <*ftp://ftp.tis.com/pub/DNSSEC/README*>. See also < *http://www.tis.com/docs/research/network/iip.html*>. TIS has received an official determination that their product is not controlled by the State Department, because it only does authentication rather than information secrecy. Probably by the time you read this, the Commerce Department will have approved their request for general export to all destinations.

I will be working with TIS and Paul Vixie to merge TIS's prototype code into the production BIND release, once the export issue is fully resolved.

A third step in DNSSEC deployment will be for top level domains (TLDs) and the root domain to accept KEY records from the owners of their subdomains, sign them, and publish them (like NS and glue records). This allows DNS clients to validate that the subdomain's signed DNS records have been issued by the entity to which the name space was delegated. For example, to verify that resource records from *usenix.org* are really from the party who was assigned the name "*usenix*" by the "*org*" domain's administrator.

Once this third step is handled, you will be able to depend on Secure DNS for production validation of name server records. Before this step, you will be able to deploy Secure DNS within your own operations, and sign the keys of zones with which you have a close relationship, but will not be able to prove to the general public that your domain records are from the officially delegated source.

Please consider installing a new release of BIND with Secure DNS support at your site. This allows you to participate in both the securing of this important piece of the infrastructure, and in using the Domain Name System to publish your public keys as new releases of popular application programs start to accept this form of keying.

The Webmaster: Web Site Memory with Cookies

by Dave Taylor
<taylor@intuitive.com>

In my previous column, I spent some time talking about the different ways you can use account/password pairs to have some history of user visits to your Web site. During the first visit, you require people to create an account, and on subsequent visits, they can pick up where they stopped by simply remembering their account and password pair.

There are, however, some major problems with that strategy, not the least of which is that you have to remember too many darn info pairs. Web sites that are offering threaded discussion spaces as an alternative to USENET newsgroups all require some sort of user validation, and as a result, in the last week alone, I've collected three new and different account/password pairs to remember. Typical solution: I use the same account name and password for all. That's not very bright and presents some potential security problems, but it's a survival mechanism.

Many times, however, all you really want to do is to be able to ascertain whether someone has been to your site before rather than track some elaborate database of information about your visitor. The first time someone visits, perhaps you'd like to have an informational page pop up that offers an "executive summary" of the site, but subsequent visits you omit that because you figure visitors know why they're on the site on return visits.

Fortunately–or unfortunately, depending on your view–Netscape has created a mechanism for visitor tracking and information compilation that is now also supported by Microsoft's Internet Explorer browser, and it goes by the whimsical name of "cookies."

What's a Cookie?

In Web terms, a cookie is a snippet of data, no larger than 4K total, that the browser stores and sends back to the appropriate server any time the user accesses the server. Each cookie, as you can see, is wedded to a specific server or, more typically, a domain. USENIX might have a cookie that its server sends when you visit its Web site that's associated with the domain "*usenix.org*."

Before you start scheming ways to hack and tweak, the highest-level domain possible for a cookie is two levels deep: you can't associate a cookie with "*.edu*", but you can have one for "*purdue.edu*" or even "*library.sail.stanford.edu*".

Cookies can be set only from within CGI scripts because they are sent along with the familiar "Content-type: text/html" preface that is required for all CGI-generated Web documents. Preface the cookie information with "Set-Cookie:". The cookies are then sent along with the rest of the client environment set to the server on each interaction.

Cookies have five parts to them:

name=value
expires=date
path=path
domain=domain_name
secure

The only field you must use is the name=value pair, which can be just about anything you want. Account=4324321 would work fine, as would HairColor=blonde. If you want to use spaces in the value field, you need to encode them as "%20" sequences, just like a URL. Multiple name=value pairs can be present; they need to be separated by semicolons and appear on the same line.

Cookies can have an optional expiration date, specified in the quasi-standard notation of "Day, DD-MMM-YY HH:MM:SS GMT." An example might be my birthday: "Saturday, 03-AUG-96" or noon GMT on the same day: "Saturday, 03-AUG-96 12:00:00 GMT." When that expiration date is reached, the cookie is purged from the client cookie storage area. (Should we call that the cookie jar?)

Within your site, you might want the cookie to be sent only to a subset of the possible pages. That's what the path variable is for; I could limit transmission of the cookie with "path=/apps", for example.

The domain allows you to specify a domain or subdomain for the cookie, such as "domain=library.usenix.org". If you specify "secure," then the cookie is sent only if the connection between the server and browser is secure (e.g., SSL).

Finally, you can specify any combination of these variables using semicolons to separate fields: "Set-Cookie: visits=1; path=/shop; domain=shop.com".

Enough nuts and bolts. Let's have some example code!

Setting and Examining Cookies

A simple shell script CGI is sufficient to set a cookie for our test site:

```
#!/bin/sh -f
echo "Set-Cookie: lastconnect=`date | sed
    's/ /%20/g'`"
echo "Content-type: text/html"
echo ""
echo '<HTML><BODY>'
echo '<CENTER>'
echo '<H1>Cookie value set</H1>'
echo '<P>'
echo "<a href=showcookie.cgi>click me to see
    new cookie</a>"
echo "</CENTER>"
echo "</BODY></HTML>"
exit 0
```

And the "showcookie" CGI script that shows the value:

```
#!/bin/sh -f
echo "Content-type: text/html"
echo ""
echo '<HTML><BODY>'
echo "<CENTER>"
echo "<H2>Cookie set to</H2>"
echo "<H1>`echo $HTTP_COOKIE | sed
    's/%20/ /g'`</H1>"
echo "</CENTER>"
echo "</BODY></HTML>"
exit 0
```

It's straightforward: invoking the first script, "setcookie.cgi", results in a screen like that shown in Figure 1 (next page), and clicking on "click me to see new cookie" on that screen produces Figure 2. Notice that we have to translate the spaces in the date output to "%20" sequences, but we can also easily unpack them when showing the cookie, both using `sed`.

These cookies are the basis of lots of interesting Web interaction, not the least of which are the many "shopping cart" type shopping sites, where you poke around and keep adding items to your basket, then at the end of your visit you "check out" and pay for all the items you have chosen. That's all done with an accumulating cookie value!

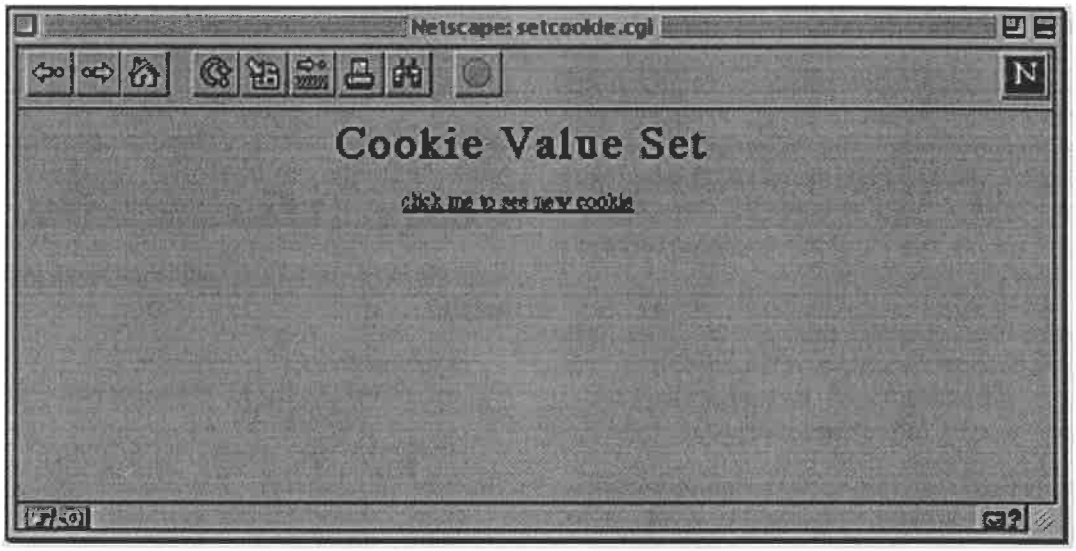


Figure 1. Setting cookies is invisible to the browser!

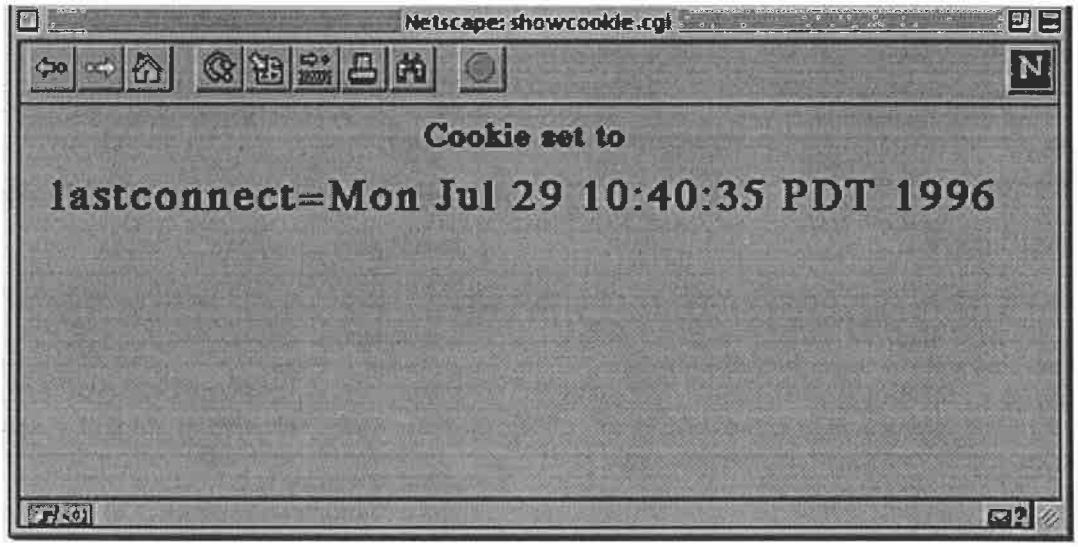


Figure 2. The cookie secretly tells us when the user (singular = correct) last visited

Warnings and Weirdnesses

Cookies don't go away and do accumulate for a site, so if I decide that instead of using "lastconnect" as my cookie for my site I want to use "lastvisit," then all the people who visited with the previous cookie value will still have that until it expires: I'll receive two values for the HTTP_COOKIE, not one.

Further, because you don't know what's going on with these mysterious critters, you have no idea what sites are storing information in your cookies, which are saved on your local disk as a Netscape or Microsoft cookie datafile. The good news is that it's a simple text file, so with a tiny bit of hacking about you can find out what's inside. On my Mac, for example, the System Folder contains another folder called "Preferences," then "Netscape f," and within that sub-sub folder there's a file called, lo and behold, "MagicCookie." I open that and here's what's inside:

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file! Do not edit.

.mcom.com TRUE / FALSE 3029529599
NETSCAPE_VERIFY
c65ffb1e,c6525d5b
.netscape.com TRUE / FALSE 946684799 NETSCAPE_ID
c65ffb1e,c6535607
ad.doubleclick.net FALSE / FALSE 942191940 IAF 3ad3c
.internet.net TRUE / FALSE 946684799 ISNER 5848192
.realmedia.com TRUE / FALSE 946684799 RMID
```

fc60ce0496ce,827423591,empty.gif
.infoseek.com TRUE / FALSE 858271949 InfoseekUserId 664FE8CC19852AD773E4FE5B6275BB83
.focalink.com TRUE / FALSE 946641600 SB_ID ads01.28742826738479109405
www.msn.com FALSE / FALSE 937396800 MC1 ID=0d7cd5667c7211cfa0c700805f482fec&CGX=0&WGX=0&GX=1&BGC=FFFFCC&TBGC=000000&VBGC=808080&UBGC=336600&PI8=on&TZ=pst&PGX=on&MCS=0&SKI1=None&SKI2=None&SKI3=None&PI1=
cgi.netscape.com FALSE / FALSE 946684799 NETSCAPE_VERIFY c65ff94b,c6d54301
.disney.com TRUE / FALSE 946684799 DISNEY 205.149.165.1092794829789835688
www.actionman.com FALSE / FALSE 946684799 INTERSE 205.158.215.13028157834042181492
.focalink.com TRUE / FALSE 946641600 SB_IMAGE 32.1.11.1::ms-ma2a.gif#29.1.1.1::ms-pcw3a.gif
.cyber-bp.or.jp TRUE / FALSE 942189160 YAMAHA-ID taylor
.adobe.com TRUE / FALSE 946684799 INTERSE 205.158.215.13015712838056924535
.imgis.com TRUE / FALSE 946684799 INTERSE ws12423838137642354
.boardwatch.com TRUE / FALSE 846000774 Userid d1taylor
.boardwatch.com TRUE / FALSE 846000774 Password fakeout!

Tons of stuff. Tons. Indeed, I don't even remember visiting some of these sites! Notice that all expiration dates have been normalized, as you'd expect, and that some sites have a ton of information they're tracking, notably the Microsoft Network (www.msn.com). Why am I not surprised?

One thing to notice here is that the boardwatch site stores my account and password pair as cookies *in the clear*, so if someone else were to pop onto my machine and look at my MagicCookie file, they'd be able to see my current password. Double plus ungood!

And Now, a Word From Your Sponsor . . .

For the last year or so I have been writing these Webmaster columns both to goad myself into exploring some of the nether regions of the Web and to share my experiences with other members of the USENIX community who seek to learn more about CGI and Web programming. I write for lots of other publications, too, most notably *InfoWorld* and *MacWEEK*, and, like those magazines, *;login:* pays its regular writers. (It came as a slight shock when I first found that out, even though I've been a USENIX member for plenty of years.)

When Ellie Young and Rob Kolstad asked me to write this column, however, I wasn't interested in getting paid for each issue, so instead we created a Web scholarship fund called "Webstar." Through its creation and award of a surprisingly large amount of money annually, I hope to help motivate student members of USENIX to create cool and interesting public service Web sites for the general public.

Please take a few minutes and flip to page XXX that talks about the Webstar scholarship award, and read what we have to say. If you're a student, apply. If you aren't, let me know what you think of the whole idea.

Some Cool Web Sites to Learn More About Cookies

http://search.netscape.com/newsref/std/cookie_spec.html
the official cookie spec from Netscape

http://www.ids.net/~oops
a pretty slick online shopping cart system w/ cookies

http://www.zdnet.com/macweek/mw_1011/gw_net_tricks.html
MacWEEK story about cookie security concerns

The Importance of Documentation

by Scott Hazen Mueller
<scott@zorch.sf-bay.org>

Simply put, documentation is the practice of recording what you have done, will do, or should do. Documentation is used for historical record keeping, to show what you have done, both as a reminder to yourself and as a way of showing those around you what you do. All of your plans for the future are also part of your documentation set. Most importantly, documentation describes how things are done in your organization and how tasks should be performed.

In this latter role, documentation is both a training tool for new administrators, and a disaster recovery tool. Documentation allows the operation to keep running even if you, the key system administrator, are run down by the proverbial bus.

Documentation and ISO 9001

The ISO 9001 series of standards deals with the consistency of an organization's quality. The gist is that the organization

determines what its processes are, documents them, and then documents that it actually follows those processes. In short, "Say what you do, and do what you say." The immediate goal is to make sure that the quality is consistent. The long-term goal for the organization is to then make use of the information gathered during the initial documentation phase and address weak points. The certification authority will return periodically and audit the organization to assess adherence to the standards and also to ensure that the organization continues to improve its processes.

If your organization is going to, or might, undergo ISO 9001 certification, you will need to document your own processes. Even if you are not going to undergo certification, however, I still recommend documenting what you do. Not only do you need the protection that documentation can provide, but it is almost impossible to improve your operation if you do not document what you are doing. When you perform the documentation exercise, you can uncover problems that were hidden. Furthermore, you can find areas that may not be actual problems, but that could still stand some work.

What To Document

It is not necessary to begin the process by documenting everything under the sun. For one, you will most likely wind up doing a poor job on everything. It would be better to do a good job on a few critical areas and leave the rest for later. Start by documenting frequently performed procedures. If you use automation for these procedures, document how to invoke the program, what the inputs should be, what errors might occur, and what the normal outputs are.

Concentrate on areas where the return is high. In addition to documenting your own procedures, document things that affect your user community, and then publicize those documents. Seek input from your users on issues that you feel might be controversial, and get their buy-in for those areas. For example, document when and why you might give users access to the root user for their own machine or for servers. Once it is written down, it is much easier to point and tell users to read it if they have questions.

If you are going for ISO 9001 certification, you will eventually need to document most things. However, you do not have to document anything that someone trained in the field can be expected to know. For example, you don't have to document how to edit the password file, because system administrators are all expected to know how to do that. You would document how to add an account because there would be activities specific to your site, such as where to put the home directory and how the user's mail delivery is controlled.

Tips on Documenting

The first thing you should have when starting a documentation process is a uniform document format. It can be quite distracting if each person in your organization formats documents differently. Agree on a single format (the ISO 9001 information comes with some suggestions), and apply it to everything and everyone. Generally speaking, you'll have some document meta-data (title, author, approver, scope, purpose, date, version) and then the document data.

Even if you are not worried about ISO 9001, it is still a good idea to put in place a document control process. It may have to run on an honor system because it's hard to keep system administrators from accessing files if they really want to, but something should be in place to help ensure that no one makes unauthorized changes to production documentation. Of course, if the master copies are kept offline, this issue vanishes, but it is much more convenient to have them online.

I recommend creating a document approval process. Even if you are the only system administrator in your organization, you still need input and review by other people before making a document official. In a large group, you can set up a section to perform technical reviews before documents are given final management approval. I have also worked with a group of users to review and approve documents that affected the user community. Once documents have been through grammatical and technical review, they should be reviewed and approved by the appropriate managers.

Documentation by itself is all well and good, but it is entirely meaningless if the documents are not conformed to. Make sure your managers understand this issue and agree to enforce conformance. In this case, ISO 9001 certification will work for you, because conformance will be required if the organization is to maintain its certification.

Using Your Documents

Your documents need to be easily accessible if they are to be useful. Hiding them away in an obscure directory is not going to help anyone, and I feel pity for the site that puts documents in a dozen different places. If you have only very basic tools (`more`, `pg`, `cd`, `ls`, 14-character file names) you probably will have to use the UNIX file system as your document management system, and create lots of directories and *Index* files.

If you have the flexibility, I highly recommend creating an internal Web. A Web is a hypertext document system using the HTTP protocol developed by CERN and used on the World Wide Web. A site can have its own private Web of documents, and they can be protected from external access in a variety of ways.

An internal Web is useful not just as an indexing system for your documentation. As more documents go online, hypertext cross-references can be added to the documents, making it much easier to find all the information related to a specific topic. For example, a server configuration might be based on the workstation configuration, so the server configuration document could contain a hypertext reference directly to the workstation configuration document.

The downside of creating a Web is that it is very easy to create a confusing and difficult-to-use document structure. My organization has been through three, and none so far has satisfied everyone. A structure that has a lot of categories gets deep quickly, and it is hard to browse documents that are related but fall into different categories. A structure that is fairly flat is somewhat easier to browse at first, but fails to scale well. I have no good answer here; I am now trying a flat index page, with documents organized by topics, and a table of contents at the beginning.

Language and Communication

by Peter H. Salus
<peter@pedant.com>

Over 30 years ago, J. C. R. Licklider astounded his readers by stating that computers were not calculation devices, but communication devices. After the Lick-inspired ARPANET and the explosive expansion of the Internet, this assertion has nearly become a platitude. Yet the computer and the networks are communication devices, and it is important for the computer professional to recognize just what communication is and how far we still have to go. A decade prior to Lick, Niko Tinbergen discussed the diversity of animal communication systems from four perspectives: mechanistic (understanding the mechanisms underlying a trait), ontogenetic (determining the environmental and genetic factors guiding development), functional (examining the trait in terms of survival and reproductive success), and phylogenetic (the evolutionary history) (*Quarterly Rev. of Biol.* [1952] 27.1-32).

Since 1952, there has been a tremendous amount of ethological work concerning communication, culminating first in the twin publications of 1977: *How Animals Communicate*, edited by Sebeok (Indiana University Press) and Thorpe's *The Behavior of Communicating* (Harvard University Press). Now there is Hauser's truly comprehensive *The Evolution of Communication* (MIT Press, 1996).

Going back further, everyone in the area owes and acknowledges a debt to Darwin's *Expression of the Emotions in Man and Animals* of 1872. I don't really want to get into historical matters (as opposed to evolutionary or developmental ones) here, but it is Darwin who pointed out, "The power of communication between members of the same tribe by means of language has been of paramount importance in the development of man; and the force of language is much aided by the expressive movements of the face and the body" (p. 354).

Tinbergen, Sebeok, Thorpe, and Hauser are all concerned with animal communication: man being the highest of the animals. But communication clearly occurs in the plants, too, each blossom sending its signal of readiness for fertilization to an assortment of insects, birds, and mammals. In fact, one might consider this the lower level of interspecies communication. It is not the lowest, for something must inform the amoeba to envelop a particle and absorb it, rather than bouncing away (or being itself eaten).

But I don't want to get into the arguments as to what is or isn't communicative ability, any more than I want to get into the morass of what is or isn't language. (In fact, here I've got to be cautious not to use an article, for what is "a language" is a different query from what is "language.") But, basically, communication requires a sender and a receiver, with a distinction between a message and its meaning. This is an interactional view of communication (much like Smith's). The message is thus the information that the signal makes available about its sender. Meaning is what the recipient makes of the signal.

Most ethologists draw a sharp line between folks and "lower" animals in terms of communicative repertoire. Scientists concur on the notion of an upper limit of about 40 signals among the animal species studied. Only P. G. B. Slater at Sussex in England seems to suggest that perhaps we just don't know enough of the nuances of animal communication. But even if we were to square that number, we'd have a small range of signals. Yet the problems of large ranges of meaning are nearly 300 years old: Leibnitz was discussing "possible-world semantics" early in the eighteenth century.

Bizarrely, although we have enormous amounts of information concerning olfactory, tactile, visual, and auditory communication in thousands of animal species, as well as the behaviors of these species in a variety of contexts, we are far from so clear where our own communicative medium–human language–is concerned. This is part of our insistence on being something special and also being part of the continuum of the "chain of being." We talk of both discontinuity and evolutionary continuity. Our languages have sound systems that are nothing like the discrete sound categories of nonhuman primates.

All of this was brought to mind by the January 1996 *CACM*, which featured "Natural Language Processing." The first article in this section, by Louise Guthrie et al., was entitled "The Role of Lexicons in Natural Language Processing". In

it the authors cite Boguraev's survey of AI/NLP researchers (1989) in which he asked how many words there really were in the vocabularies of their systems: The average was 36. The range was nearly identical to that mentioned by animal communication researchers. Guthrie et al. are interested in moving this impoverished vocabulary several orders of magnitude and do an excellent job of discussing computational lexicons, the taxonomies inherent in dictionary definitions, and methods for construction of a lexical database.

One of Guthrie's co-authors, James Pustejovsky, has published *The Generative Lexicon* (MIT Press, 1996). Here Pustejovsky presents a theory of lexical semantics that attempts to cope with the problems attendant on multiple word meanings. The result is a treatment of event semantics and polysemy that may have tight logical cohesion, but bears no resemblance whatsoever to what psychologists and linguists have learned over the past 25 years. E. and H. Clark; K. Forster; Smith, Shoben, and Rips; Rosch; and the ethologists are all missing.

To me, this is the major problem that nearly all of the NLP work has: it may mechanically execute algorithmic substitutions, but it doesn't come close to what each of us does with the 2 kg of (largely) fat and water in our heads.

In 1983, I published a long article "A Realistic View of the Mental Lexicon" in *Semiotica* (vol. 43, pp. 337-365). In it I listed a number of things that each of us can do with our mental lexicons. Among other things, I remarked that the following were "more-or-less normal situations":

- Someone asks if you know a five-syllable word for "unending" and you respond "interminable"
- A child asks what "technology" means and you define the term
- You list some animals from "aardvark" to "zebra," some flowers from "aster" to "zinnia," and some countries from "Afghanistan" to "Zaire"
- You solve a crossword puzzle
- You speak and comprehend speech
- Someone makes a pun and you laugh at it
- You hear something that starts a chain of rhyming or alliterating words running through your head
- You interpret a metaphor; you use an idiom.

In other words, you accessed information about words that you know, in a variety of ways: by definition, by sound, by syllabification, by spelling, etc. Each type of access requires that there be a special kind of storage, a specific kind of "marking" that will enable us to do a "lookup."

But if we think of this when we look at research like that contained in the Guthrie et al. article, we'll immediately notice what's wrong: we're not even talking about doing the same sorts of things. Guthrie et al. deftly switch from using lexicon to using dictionary, but it doesn't matter. They state, "The purpose of dictionaries is to provide a wide range of information about words–etymology, pronunciation, stress, morphology, syntax, register–to give definitions of senses of words, and, in so doing, to supply knowledge not just about language, but about the world itself" (p. 64). They then move on to online dictionaries and to several research programs.

I received the following via email:

> Two hillbillies meet walking along a road in the Ozarks.
> One says:
> "What's in the sack?
> "Chickens."
> "If I guess how many, can I have one?"
> "You can have both."
> Pause.
> "Five?"

It is hard to understand what my online lexicon will make of this. Even if the syntax is comprehensible, and it knows that "both" means "two," what will it understand "about the world itself" by the guess "Five"? Will it put into its lexicon under "hillbilly, -ies" the fact that the word implies a certain lack of awareness? Will it realize that the site being in the Ozarks and that this year being an election year render this a political joke? I see all of this as part of "the world itself."

We can go a great deal further in real-world understanding. Take the concept of "opposite." Most of the readers of this would be pretty good at telling me the opposite of a given lexical item: up/down, in/out, clerical/lay, remember/forget, etc. But what about "sweet"? Sweet/sour, sweet/salt, sweet/dry, sweet/nasty all come to mind; dry as in vermouth, sweet/nasty in behavior or temper. Where will such contexts occur in our online lexicon, or (in fact) how will our machine recognize the difference between a "pine cone" and an "ice-cream cone" (a question brought up a decade ago by Mike Lesk at the Toronto SIGDOC)? I still don't know.

In 1984, I was working on a grammar-checker project. One of the problems I had (and which I never solved) was how to "explain" to the machine that although verbs following (s)he generally end in -s (she reads, he buys), there are a few places that this isn't true, e.g.:

> I wrote to Dennis and asked that he send me two copies.

The embedding of a sentence within another is the determinant. I never got the machine to look back far enough to recognize the embedding. It just told me the sentence was ungrammatical. I have since ascertained that most nine-year-olds understand and respond appropriately to this kind of embedding. Perhaps our grammatical-lexical computer programs are below that comprehension level.

I have not written all this to mock NLP programs or research: far from it. What bothers me is that we forge ahead with our pragmatic attempts at this while we comprehend so little about what we do.

Another part of all this comes from my recognition of the richness and complexity of natural language while I was reading Gosling, Joy and Steele's *The Java Language Specification*. Java is a relatively rich language; it is strongly typed; it is at a rather high level. But the 800+ pages here are really fairly complete. I have several hundred volumes on the English language. They are not complete. They may never be.

Using Java

by Rik Farrow
<rik@spirit.com>

Often, in this community, suggesting that something should be done is tantamount to volunteering to do it. Well, perhaps this is also true of committees that get things done. In my case, it means that this is the initial article in a series about using Java. In the future, we expect that other writers/programmers will also contribute to this feature.

This article is not a beginning tutorial. While still covering some of the basics for getting started, it assumes that you know something about Java. I suggest *Exploring Java* (Niemeyer and Peck, O'Reilly and Associates), some of the other books mentioned in *;login:* reviews, or the online documentation available from *<http://www.javasoft.com>* for getting started with Java.

Java has survived well beyond the hype stage. Its initial popularity followed on the coattails of the HotJava World Wide Web browser. Soon, Netscape had decided to support Java applets, and surprisingly shortly after that, even Microsoft licensed Java. HP and IBM are working on Java support (Java will be supported within OS/2 and AIX), and a port has been completed to Linux and many other PC UNIX-clones. In a future feature, I will list pointers to various free versions as they become available. The USENIX Web site would be another good place for pointers.

For now, the place to get a version of Java is *<ftp.javasoft.com>*. This server is often busy, and has a limit of 150 simultaneous connections and an annoying short timeout. If you don't get in, a list of mirror sites is listed.

What you want to download is the Java Development Kit (version 1.0.2 at this time). The JDK is free, and versions for Solaris 2.x (both SPARC and x86), Windows 95 or NT, and MacOS 7.5.x are available from Sun. To find versions for other operating systems, try a search engine with the letters jdk.

The Tools

The JDK unpacks into several directories. HotJava is missing from 1.0.2. The previous version, 1.0.1, included HotJava, which is useful for learning how to load Java classes dynamically. I plan to show examples of loading classes over the network in a future article, but mention this now in case you haven't already discarded a 1.0.1 version of the JDK.

You will want to add the `java/bin` directory to your `PATH`. The `bin` directory contains the executables and the shared objects (which contain references to native libraries). You will also want to create a new variable named `CLASSPATH`, which includes the `java/lib` directory, any directory you will keep your own classes in, and `dot`, the current working directory for development (non-developers shouldn't include `dot` in their `CLASSPATH`s). For the Windows 95 impaired, such as myself, I'd like to point out that the separator for `PATH` and `CLASSPATH` elements is the semicolon, not the colon.

The `javac` program compiles Java classes. Each class (or perhaps several classes) is contained in a file with the same name as the public class in that file and the `.java` extension. A successful compilation results in a `.class` file for each class defined in the `.java` source file.

When you compile using `javac`, you must include the `.java` extension in the filename. Common mistakes include not using the same filename as the public class defined in the `.java` file, and not having `CLASSPATH` correctly defined.

In the JDK, the Java compiler is actually a Java application. This makes it easier to port Java to other environments, but at the cost of some performance. I have been told that Microsoft's J++ compiles Java source much faster, and I suspect that Microsoft, unconcerned with portability issues, has written their compiler in native code. Other development toolkit vendors will likely follow suit, not that compilation takes terribly long on fast systems anyway.

The `java` program implements the Java Virtual Machine. The Java Virtual Machine turns Java bytecodes into native machine code and interfaces with the native operating system. In some implementations of the JVM, you can optionally produce "just-in-time" compiled code, which means that the first pass through the interpreter results in machine code which speeds up subsequent execution. For a lot of what is done with Java, such as user interfaces, a "just-in-time" compiler is overkill. Remember that this compiler is part of the runtime environment, not the compiler which produces the classes.

The JDK also includes an `appletviewer`, which can view remote or local files (using either the file or HTTP style URL's) and interpret <applet> tags. Of course, both Netscape

Navigator and Microsoft Explorer also support the <applet> tag, but the `appletviewer` is still nice for testing purposes.

Applications vs. Applets

The end product of compiling Java source code is always one or more class files. If the class is intended for standalone use (an application), it must have a class method named `main()`. In Java, class methods (those associated with the class rather than instances of a class) are labeled with the keyword `static`. The JVM, started via the java command, creates an instance of the class provided as its argument, starts a thread for this class instance, then invokes the `main()` method for this class with an array of strings as an argument.

Applets work a little differently. The browser or appletviewer loads the class file when it interprets the <applet> tag with a special class loader (which treats classes from the network differently than classes loaded locally). After loading the class file, browsers go through another step, byte code verification, under the assumption that remotely created code may be malicious or not compiled with standard tools, leading to overwriting the stack, crashing the browser, and other mischief. If the class passes, an instance of the class is created, and a thread started for this class.

Unlike standalone applications, a browser does not call `main()`, but instead makes a series of calls to the applet's methods. The `init()` method is called once, after the thread has been created, followed by `start()`. The `start()` method is recalled anytime the Web document containing the <applet> tag is revisited, and the `stop()` method is called when another Web document is loaded. The `paint()` method gets called when the applet's Panel gets exposed, for example by scrolling up or down through the Web document. The `paint()` method can also be called directly, or via `repaint()` or `update()` calls within the applet.

Going Both Ways

Examples often make things clearer, but I take a risky approach in the one following. Clock.java contains a class which can be used as a standalone application or as an applet, and works similarly either way. Through using this example I plan to compare and contrast the way applications and applets work, and how they are written.

The source file begins with an <applet> tag wrapped in a C-style comment. This is a trick I learned from the Patrick Naughton book (*The Java Handbook,* Osbourne McGraw-Hill, 1996). If you include an <applet> tag in a comment, you can then test the applet within the `appletviewer` without writing a separate Web document (HTML file). For example,

```
appletviewer file://home/rik/Clock.java
```

will start an `appletviewer`, which will in turn load the Clock.class file if it is found in the same directory (or you can specify a different directory in the <applet> tag).

```
/*<applet code=Clock.class height=110
            width=200>
</applet>
*/

import java.awt.*;
import java.util.Date;
```

The `import` keywords tell the compiler where to look for non-locally defined methods or classes. Unlike C or C++, there are no `#include` statements, and importing a Class doesn't include its bytecode, but only makes the compiler aware of it. Using an asterisk means that all classes in that package (or directory) will be included, which can slow down compilation.

The `class` keyword marks the beginning of the class definition. There can be at most one public class for source file. This example includes a second, non-public class later on. The instance variables will be available to each instance (created using the `new` keyword and the Class name). Static, or class variables, exist as part of the class, and are available before any instances have been created.

```
public class Clock extends
          java.applet.Applet
    implements Runnable {

// Instance variables
    // Handle for Thread object
    Thread aThread;
    // Handle for Date object
    Date date;
    // Handle for window (Frame object)
    Frame frame;
    // Handle for drawing graphics
    Graphics graphics;

// A class variable to distinguish between
// Applet and the standalone version
    private static boolean isApplet = true;
```

A `public` method with the same name as the class being defined is called a *constructor.* The constructor is always called when a new instance is created. The first statement within a constructor, which may be implicit (as in this example), is a call to `super()`, the constructor for the superclass. The superclass, in turn, also calls its `super()`, and so on, up to the root class, Object.

In a constructor, you usually include initialization for the instance just created. Here, I must distinguish between when the constructor is called from `main()`, the standalone ver-

sion, or from within an applet, using the class variable `isApplet`. Equivalent initialization for the standalone version is performed in the constructor for MakeWindow.

The Panel instance is created to hold the two Buttons, and then the Panel `p` is added to Panel which the browser has created for the applet. Applets are subclasses of Panel, so the method invocation `add("North", p)` applies to the new instance of Clock, a subclass or Applet, which is a subclass of Panel. The Panel's `add` method handles adding the Panel as another Component.

```
public Clock() {
    if (isApplet) {
    // Standalone version uses MakeWindow()
        Panel p = new Panel();
        p.add(new Button("Start"));
        p.add(new Button("Stop"));
        add("North", p);
    }
}
```

The next three methods will get called by the browser automatically after loading the applet. The `start()` method is also called directly from `main()` in the standalone version, and by the event handling routines to start the thread. The thread is created in `start()` with "this", a reference to the instance through which the method was invoked, making the instance of the Applet the target of the thread.

```
public void init(){
    graphics = getGraphics();
    date = new Date();
    }

// start() is called when applet's panel
// becomes visible

public void start() {
    if (aThread == null) {
        aThread = new Thread(this);
        aThread.start();
        }
    }

// stop() is called when the appletviewer
// changes pages

public void stop() {
    if (aThread != null) {
        aThread.stop();
        aThread = null;
        }
    }
```

Threads get started or stopped via calls to their `start()` and `stop()` methods (they can also be suspended or resumed). Starting a thread causes its `run()` method to be invoked. When the `run()` method exits, the thread dies. Here, the applet's `start()` and `stop()` methods (above) call the thread's `start()` and `stop()` methods to start the thread, or destroy it.

```
public void run() {
  while (true) {
    date = new Date();
    paint(graphics);
    try {Thread.sleep(1000); }
    catch (InterruptedException e) {};
  }
}
```

The `paint()` method gets called when the applet needs to be displayed, or when called explicitly from the `run()` method. The Graphics object provides an interface to the underlying toolkit used to display graphics in the local windowing environment (another topic in itself). Here, I use `clearRect()` to erase the region where the date will be displayed, and `drawString()` to display the date. Date objects contain the date as a long (64 bit value representing the number of milliseconds since the Epoch), and the `toString()` method converts this to a more readable String.

```
public void paint(Graphics g) {
    g.clearRect(25,25,150,50);
    g.drawString(date.toString(), 25, 60);
}
```

Because I have chosen to make this both an application and an applet, I needed two ways of handling user events. In the applet, the `action()` method gets called by the `handleEvent()` method of the Panel with two arguments anytime either of the two buttons, Start and Stop, get selected. The applet's `start()` or `stop()` methods get called, starting or stopping the thread.

```
public boolean action(Event e, Object arg) {
    if (e.target instanceof Button) {
        if (((String)arg).equals("Stop"))
                stop();
        else start();
        return true;
        }
    return false;
}
```

For the standalone version, we need a couple of things. First, we must have a `main()` method, defined as `static` (a class method). I set the class variable `isApplet` to false before creating the Clock object so the constructor will not create the Panel and two buttons. Then a MakeWindow object is created, which creates a standalone window (Frame object), and adds the Panel with the Start and Stop buttons.

Notice that most of the methods and variable references in `main()` are preceded with an object handle. In Java, all references to non-static methods and variables are associated with an object. When no object is specified, `this`, the instance the method was called through, is implied.

```
public static void main(String args[]) {
    isApplet = false;
    Clock clock = new Clock();
    clock.frame = new MakeWindow(
                "Clock Window", clock);
    clock.graphics =
                clock.frame.getGraphics();
    clock.start();
    }
}

// A second helper class; could be in a
// separate file
class MakeWindow extends Frame {

// Instance variable
Clock clock;

// Constructor for new MakeWindow object
public MakeWindow(String name, Clock
                  applet) {
    super(name);
    clock = applet;
    Panel p = new Panel();
    p.add(new Button("Start"));
    p.add(new Button("Stop"));
    add("North", p);
    resize(200, 110);
    show();
    }
```

The MakeWindow instance has its own Frame, and must also have its own event handler. Instead of using `action()`, the `handleEvent()` method for the Frame class is overridden. The `action()` method would handle the Button events, but there is no "shortcut" method for window destroy events (choosing Close or Quit from the Window's menu). I was a bit surprised to find that the toolkit handles events such as resizing or minimizing Java created Frames, but not quitting them. You must capture `WINDOW_DESTROY` events, and do the work yourself.

```
public boolean handleEvent(Event e) {
  if (e.id == Event.ACTION_EVENT) {
    if (e.target instanceof Button) {
      String s =
        ((Button)e.target).getLabel();
      if (s.equals("Stop"))clock.stop();
      else clock.start();
    }
  }
  else
  if (e.id == Event.WINDOW_DESTROY) {
    this.hide();
    System.exit(0);
    }
  else
    super.handleEvent(e);
  return true;
  }
}
```

Future Events

There is much to learn about in Java. Although this example is a bit long, I hope to provide some shorter and yet compelling examples in future features, along with contributions from other Java programmers. If you have never looked closely at Java, this example may do more harm than good. It only hints at the power you get from using some of the predefined classes which come as part of the core Java API. The ability to use ready-made classes is where the real future of Java is, and the real future for the programmers and companies who will be designing and writing those classes and packages.

WEBSTAR AWARD COMPETITION

Do you know anybody in the K-12 age group who has developed a Web site? If so, they may be eligible for the first WebStar Award to be presented at the Annual USENIX Technical Conference in Anaheim in January.

The WebStar Award, announced by USENIX and Intuitive Systems, will be given for the best public service Web site developed by an individual or group in the K-12 age group. The Award is aimed at increasing awareness of public service Web sites and promoting the remarkable abilities of youth in this dynamic new publishing arena. If you know anybody who falls into this category, please tell them about the WebStar award, and point them to the Web site: *<http://www.usenix.org/webstar>*.
There's an entry form and more information about the award.

The WebStar Award consists of a cash prize of $1,000.00 plus travel to the Annual Technical Conference in Anaheim to receive the award for the lead webster and a parent.

Awards will be granted based on a combination of interface design, friendliness, comprehensiveness, accuracy, and the inherent value of the site to the online community.

Intuitive Systems of Redwood City, California, is an interface design firm that focuses on enhancing the user experience and creating navigable spaces. The company is well known as the creator of the Internet Mall (tm) at *<http://www.internet-mall.com/>* and a wide variety of other Web sites. President Dave Taylor has been involved with USENIX for over a decade, and has a Masters in Educational Computing from Purdue University.

For more information, contact Dave via email at *<taylor@intuitive.com>* or Zanna Knight at *<zanna@usenix.org>*.

An Update on Standards Relevant to USENIX Members

by Nicholas M. Stoughton
USENIX Standards Report Editor
<nick@usenix.org>.

The following reports are published in this column:

- A Report from the Chair

See also Nick Stoughton's "Are Standards Worth the Effort" on page XXX

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send any comments you might have to *<nick@usenix.org>.*

A Report from the Chair

by Lowell Johnson
<Lowellrsvl.unisys.co>

A "Fast-Track" Standards Process

The Portable Applications Standards Committee (PASC), in close cooperation with the IEEE Standards Department, is continuing to investigate ways to improve the general process of producing standards, as well as developing special purpose processes for situations that are not well served by the normal full-blown standards development process. This article will outline a "Fast-Track" process that would enable achieving full standard status in approximately six months, yet follow all IEEE formal procedures, with two small deviations. This process is currently only proposed for use in PASC, but other groups could adopt it if we are successful.

The basic premise is that an organization comes to PASC with a specification document that is already complete. This specification would then be taken through an abbreviated balloting process with a simple YES or NO vote, with no binding objections allowed, and no recirculations required. Editorial comments would be considered, but suggestions for changes or enhancements would only be accepted for possible future consideration. If there are serious problems, the balloter would vote NO with the normal appropriate rationale. If 75% approval is achieved, the Fast-Track succeeds; if not, it fails and the project is withdrawn.

The following is a brief summary of a hypothetical case, which will be followed by a more complete process description. The example was chosen to start in July, but it could be at any one of our quarterly meetings. Remember that the document must be well formed when it is initially submitted and should not be an overly large document since we will be pushing the bounds of the balloting process. This process is best suited for amendments and small, stand- alone standards.

1. July: A completed specification is brought to PASC and accepted for Fast-Track consideration.
2. July: IEEE is immediately requested to form a ballot group.
3. July: A Project Authorization Request (PAR) is also submitted to the IEEE Standards Board for approval.
4. August: Ballot group formation takes place.
5. September: The Standards Board approves the PAR.
6. September: The 30 day ballot is immediately sent out.
7. October: The balloting period.
8. November: Ballot results are submitted to the Standards Board for approval.
9. December: Standards Board approves (assuming the ballot passes).

The following detailed explanation of each step will also note in particular where this deviates from normal standards development processes. The assumption made is that each step is successful, because if it is not, the process either stops

completely, or it expands into a full-blown standards development process (with a revised PAR).

1. PASC is asked to consider a specification that has been submitted for Fast-Track approval. PASC normally requires that a PAR be created and submitted well in advance of a scheduled meeting. This allows time (for example) to assess the impact on resources to write the standard, evaluate the viability of the technology, and determine if there is sufficient expertise to complete the work. However, in some circumstances, everything could be done at the first meeting, if necessary, since most of these issues would be moot because a complete specification is in hand.
   It would still be desirable to have the document in PASC's hands well before the meeting for evaluation. This should usually be possible since the submitting organization, knowing they were going to seek Fast-Track approval, could submit a draft document (if necessary) months before the meeting.
   Approval for sponsorship would be based on the assumption that PASC would be presented with a complete specification, which was produced by major "players" in the field, and for which there was already substantial consensus. PASC would not sponsor a ballot where work was being done by another group, or where the document was obviously not complete or professionally produced. PASC may also simply choose not to sponsor work they felt was not appropriate for standardization or not within its scope.

2. IEEE is immediately requested to form a ballot group for this project. This is the first small deviation from normal IEEE procedure, since the normal requirement is that there be an approved PAR before ballot group formation begins. However, there is very little down-side cost in this deviation. If the PAR is not approved, the group is dissolved, and the IEEE has wasted only some paper and postage. There is also a requirement that a document enter ballot less than 90 days after the ballot group has been formed, but since we have the document in hand, this is not a concern.

3. The PAR is sent to the New Standards Committee (NesCom) and the Standards Board for approval at their next meeting (September in this example).

4. The following month (August) is the 30 day open enrollment period for this ballot group.

5. NesCom and the Standards Board approve the PAR at their September meeting.

6. The address labels, ballots, etc. have already been produced so the ballot can be sent out immediately. Even with the allowed time for surface mail to arrive, the ballot will be in the hands of the balloters by the end of the month. Once we move to electronic balloting, this schedule, which now seems very tight, will become easier to meet.

7. The next month (October) is the balloting period. At the end of this step, the second deviation from normal IEEE procedure occurs. Normally, even if the required 75% approval is achieved, all ballots must be recirculated to the entire ballot group. This allows any positive ballots to change to negative after seeing the other ballots, which may have brought up issues they had not considered.
   By definition, this is an up-down ballot with no binding objections allowed and no recirculations. Even in the normal standards development process, this last recirculation rarely generates significant changes in the ballots. However, it should be emphasized that if significant issues are brought up, both the original developer of the specification and PASC always have the option of dropping out of the Fast-Track process.

8. The balloting results are sent to RevCom (the final standards Review Committee) and the Standards Board for approval by the required deadline (in November in this example).

9. RevCom and the Standards Board approve the standard.

Up to this point a potential problem has been ignored that could significantly delay the publication of the standard, even after it is approved. The document must be in a format that is consistent with the general IEEE document format. If it is not, someone must spend significant time to edit the document into the correct format. For example, one of the major consortia writes their specifications using the word "should" for normative statements, while the IEEE requires the use of the word "shall." This may seem minor, but such a re-editing could significantly delay publication of the standard.

However, most major corporations and consortia know what the IEEE format requires. If they plan to pursue a Fast-Track process, they can either develop their documents in the required format, or modify them before submission. Re-editing, either before or after balloting would appear to be the more costly choice, so development in the approved format should be encouraged.

Although this whole process may seem almost intuitive, it requires very tight control of schedules, a quality document at the outset, and the forbearance of the IEEE Standards Department. For these reasons, the Fast-Track process will never account for more than a small minority of all sponsored standards. However, if we can make this process work for the cases where Fast-Track approval is important, we will have achieved something important.

The Bookworm

by Peter H. Salus
<peter@pedant.com>

I have been Java-ed to death this year. More than a dozen volumes have arrived over the last two months. I found several of them worth mentioning. But one of them is outstanding.

Best Beans

The very best beans this month can be ground from the contents of the massive *Java Language Specification*. For those of us who are interested in languages and how they are built, this is an exemplary job. I cannot laud what Gosling, Joy, and Steele have done here sufficiently. Just buy it and consult it. But don't expect it to be easy. No book that begins with "Java is a general-purpose, concurrent, class-based, object-oriented language" is going to be as accessible as *David Copperfield*. Next best are the two volumes of the API, which are straight from the "horse's mouth."

In *On to Java* (which is not about the exploration of Indonesia), Winston and Narasimhan present a clear and succinct introduction to the language. I enjoyed reading it. Also interesting is Niemeyer and Peck's *Exploring Java*. In the competent way we have come to expect from O'Reilly's authors, they set out a comprehensive introduction in under 400 pages. *Java by Example* purports to be for "intermediate and experienced programmers." "Beginning to intermediate" would have been more like it. But it is a good instance of how you can teach by introducing the student to real code that works. The accompanying CD-ROM includes code for the Java Developer's Kit (for Solaris, Windows NT and Windows95, and Beta for Mac7.5; large letters proclaim "Windows 3.1 IS NOT SUPPORTED"), Cafe Lite, and other stuff.

Making Sense of Java is a very different sort of book, and it should be a useful one. This is the book for you to toss at your director, vice president, or whoever doesn't know what Java is, what it does, and why you intend to use it. In 150 well-written pages, the authors have explained the concepts, features, benefits, limitations, etc., of Java. A tip of my Java propeller-beanie.

IPC

There are under 30 pages on IPC in Leffler et al.; that's gained about 100 pages in the 4.4BSD book. Gray has produced a good book, well worth your time. It told me a lot about interprocess communications. I like the table of the first 33 signals a lot.

HTML

I admit that I'm bored with HTML, but two of the recent items are worth mentioning. One is Tennant's *Practical HTML*. This is a self-paced tutorial that is most notable for its small compass and graspability. I think Tennant has dramatically reduced the number of tags to learn and introduced material in such a way that wanna-bees can become users. It comes with both Windows and Mac diskettes. I actually found someone to let me look at the Windows version. If you have learners around, this is something to steer them to.

At the other end of the weight scale is *JavaScript Sourcebook*. JavaScript is the built-in HTML scripting language supported by a whole bunch of systems. Over

Books Reviewed in this Column:

James Gosling, Bill Joy, and Guy Steele, The Java Language Specification. Reading, MA: Addison-Wesley, 1996. ISBN 0-201-63451-1. Pp. 825.

James Gosling, Frank Yellin, and The Java Team, The Java Application Programming Interface. Reading, MA: Addison-Wesley, 1996. Vol 1, ISBN 0-201-63453-8. Pp. 494. Vol. 2, ISBN 0-201-63459-7. Pp. 406.

Patrick Henry Winston and Sundar Narasimhan, On to Java. Reading, MA: Addison-Wesley, 1996. ISBN 0-201-49826-X. Pp. 329.

Patrick Niemeyer and Joshua Peck, Exploring Java. Sebastopol, CA: O'Reilly & Associates, 1996. ISBN 1-56592-184-4. Pp. 407.

Jerry R. Jackson and Alan L. McClellan, Java by Example. Upper Saddle River, NJ: Prentice Hall, 1996. ISBN 0-13-565763-6. Pp. 345 + CD-ROM.

Bruce Simpson et al., Making Sense of Java. Greenwich, CT: Manning, 1996 (distributed by Prentice Hall). Manning ISBN 1-884777-24-4; P-H ISBN 0-13-263294-2. Pp. 159.

John Shapley Gray, Interprocess Communications in UNIX. Upper Saddle River, NJ: Prentice Hall, 1996. ISBN 0-13-186891-8. Pp. 364

Roy Tennant, Practical HTML. Internet Workshop Series 6. Berkeley, CA: Library Solutions, 1996. ISBN 1-882208-19-6. Pp. 105 + 2 diskettes.

Gordon McComb, JavaScript Sourcebook. New York: Wiley, 1996. ISBN 0-471-16185-3. Pp. 725 + CD-ROM.

Nancy J. Yeager and Robert E. McGrath, Web Server Technology. San Francisco: Morgan Kaufman, 1996. ISBN 1-55860-376-X. Pp. 407.

Brad Cox, Superdistribution. Reading, MA: Addison-Wesley, 1996. ISBN 0-201-50208-9. Pp. 205.

Java Class Reference Package. Seattle, WA: SSC, 1996. 2 folding reference cards.

700 pages and a CD-ROM with a library of applications and routines. This is a good book, but I think I prefer Tennant's terseness.

Web Servers

I was quite pleasantly surprised by Yeager and McGrath. Their *Web Server Technology* is complete and useful. I especially like their chapter on digital commerce, which includes a very fine section on integrating cryptographic algorithms into the Web. They also have some excellent pages on performance, a topic omitted from too many other books.

Brief Notes

I also enjoyed Brad Cox' *Superdistribution.* This is a brief tract on the concept of use rights, as opposed to copyrights. This is an important topic and one we'll all be considering in the future.

I get confused easily. I used the FSF's emacs reference card at least once last week. As a result, I was fascinated by the two Java "Class Reference" cards from SSC. One is for applet, `awt`, and `util`; the other is for `lang`, `io`, and `net`.

I guess valuable, readable, and useful are this month's magic words.

The Java Programming Language

by Ken Arnold and James Gosling, Addison-Wesley, 1996, ISBN 0-201-63455-4, 333 pp.

Reviewed by George W. Leach
<gwl1@gte.com>

If you have been waiting for just the right book on Java to appear on the market and are a big fan of *The C Programming Language* (Kernighan and Ritchie, Prentice-Hall, 2nd ed., 1988) or *The C++ Programming Language* (Bjorn Stroustrup, Addison-Wesley, 2nd ed., 1991), then this is the book for you. Jim Gosling is the creator of Java at Sun, and there is no one more qualified to write this book.

Chapter 1 provides a quick tour of the Java language, briefly touching upon all its major aspects, including the ubiquitous Hello, World program, but as a Java class. Chapter 2 discusses classes and objects, while Chapter 3 tackles extending classes and Chapter 4 presents the topic of interfaces. This is a bit of a departure from the format of the earlier books on C and C++, both of which dive right into the beginning elements of the respective languages in Chapter 2. *The Java Programming Language* defers treatment of the basic elements of the language until Chapter 5 ("Tokens, Operators and Expressions"). But that's OK. It provides a broader overview of the capabilities of the language than a quick tour prior to jumping into the details of each part of the language in isolation.

The next several chapters deal with control flow constructs, exceptions, strings, threads, packages, and the I/O package. Once these chapters have been mastered, the reader has learned most of the language. Chapter 12 covers the standard utilities that come with Java, such as vector, date, hashtable, dictionary, and bitset classes. Chapter 13 discusses programming with types, which discusses the primitive types of Java such as boolean, char, byte, and int and their corresponding wrapper classes including Boolean, Character, Number, Integer, Long, Float, and Double. Also discussed is the special Class class and Java runtime class loading. System programming is the topic of the final chapter. Covered here are access to the Java runtime environment and the underlying operating system services, such as I/O, memory management, and more.

Appendix A covers native methods that allow Java to interface to existing C and C++ code. The Java runtime exceptions are covered in Appendix B. And Appendix C provides tables of keywords, special characters, operator precedence, and Unicode.

This book does not provide a language reference, as is found in K&R and Stroustrup. Nor is the Abstract Windowing Toolkit (AWT) or the Java Developers Kit (JDK) discussed here. However, other books in the Java series will provide a specification for the Java Language, an API reference, tutorial introduction to programming with Java, and more. See *<http://www.javasoft.com/Series/>* for more information on this book and others that will be forthcoming in this series.

The Java Programming Language, like the earlier books on C and C++, is not for novices. It is for the serious programmer who needs a book that can serve not only as an introduction to the language, but also as a desktop reference for many years to come. Put this one on your bookshelf next to K&R and Stroustrup.

Exploring Java

by Patrick Niemeyer and Joshua Peck, O'Reilly $ Associates, 1996. ISBN 1-56592-184-4, 391 pp., 16 pp. index, $24.95

Reviewed by Rik Farrow
<rik@crow.spirit.com>

I first got a look at *Exploring Java*, the new O'Reilly tutorial, while at the USENIX security symposium. What I saw looked interesting, but I wasn't able to really focus on the book because an attractive young woman from a university kept asking me questions. After I finally escaped, I realized I had had my first USENIX groupie experience. I can be really slow sometimes.

The O'Reilly folks finally got around to sending me a review copy (they were unusually slow this time too), and I was able

to read a good part of the book while traveling to a Java class I taught in Denmark. What I saw pleased me.

Unlike *Java in a Nutshell*, *Exploring Java* is a tutorial that moves comfortably. Rather than taking a nuts-and-bolts approach, after a little history and some feature descriptions, the authors get you interested with simple examples–several versions of the classic Hello, World. Before the end of the second chapter, you have created applets which include simple animation and a GUI component. I found this approach better than what I remember seeing in other Java books because it engages the reader faster.

The authors have provided a true tutorial, which includes both examples and complete explanations of many aspects of Java. Most books on Java, including the Addison-Wesley Java series, don't go into enough detail about many of the classes in Java. *Exploring Java* does not skimp on explanations, which I like.

Not only are the explanations good, but they also seem carefully researched, with attention to important side topics. For example, the authors explain the relationship between Java components, such as Buttons or Scrollbars, and their native Toolkit counterparts. *Exploring Java* also spends more time than I have seen elsewhere to expanding on how the SecurityManager class works, where it fits in, and how to use it.

After the introductory chapters, topics include Threads, I/O (Streams and Files), utility classes, network programming, the Abstract Windowing Toolkit, and using images. I occasionally found what I thought were mistakes, but a good part of the time the authors were right. For example, a Thread can change its own priority to one greater than its parent Thread's priority. But they wrote that random values for Floats and Doubles range from -1.0 to +1.0 (the values are always positive in the versions of the JDK I have used).

I already recommended *Java in a Nutshell* as the essential deskside resource for writing Java applications. *Exploring Java* is a very good tutorial, and you will probably get much out of it even if you already know a lot about Java.

Denmark

And how was Denmark? I had fun with the class, except that it was too hot (31 Celsius, or about 86 Fahrenheit). This much heat is unusual in Denmark so late in the summer (week of August 18), so I was unprepared. On Sunday, people had thronged to the beaches, where my host pointed out that I needed no bathing suit if I wanted to swim. The stream of lovely bicyclists passing by already had my head swimming. The Danes are a friendly people, English appears to be the second language (even though German is really second), and I always enjoy visiting there. I recommend not passing up any chance you have to visit.

HTML: The Definitive Guide

by Chuck Musciano and Bill Kennedy, O'Reilly & Associates, 1996, ISBN 1-56592-175-5, 385 pp., $27.95

Reviewed by George W. Leach
<gwl1@gte.com>

The computer book publishing industry has benefited from the rising popularity of the Internet and World Wide Web. Just take a stroll down the aisles of your favorite bookstore and you won't believe the number of Internet-related titles that have appeared over the past six months or so. Unfortunately, many of these books are not very good. Others are out of date the moment they are printed or barely cover the subject matter. Fortunately, the dependable people at O'Reilly & Associates have come to the rescue, at least for the HyperText Markup Language (HTML) author/programmer with this new book by Chuck Musciano and Bill Kennedy.

HTML: The Definitive Guide provides the reader with detailed coverage of the syntax, semantics, and usage of the current HTML standard, HTML 2.0. Also included are discussions of popular vendor extensions such as tables, frames, Java, and other HTML tags. Throughout the book, the authors point out what works and doesn't work for the most popular browsers, including Netscape Navigator, Microsoft Internet Explorer, and Mosaic. In addition, the authors cover issues relating to style. For those not keeping current with the World Wide Web Consortium (W3C) standardization efforts, the derailed HTML 3.0 effort is discussed in a section of the preface.

The book is well organized, with chapters devoted to an overview of HTML and the World Wide Web, an HTML quick start, HTML document structure, textual elements of HTML, images and multimedia, hyperlinks, lists, forms, tables, frames, and some vendor features. Each chapter provides both a tutorial introduction to the topic at hand and an extensive reference section on each feature, making this both useful for learning about HTML and as a desk reference later on.

Appendix A provides a grammar for HTML to assist those not versed in the world of SGML and the corresponding Document Type Definition (DTD), which formally defines HTML 2.0. Other appendices cover the HTML 2.0 DTD, present an HTML tag quick reference, list character entities for HTML, and color names and their corresponding values. Also contained in the back of the book is a handy pull-out HTML quick reference card. Unfortunately, the quick reference card does not differentiate between HTML 2.0 standard tags and vendor extensions like the book does.

Sprinkled throughout the book are quite a few HTML programming guidelines to aid in developing effective Web pages. An appendix that collected these guidelines in a single, easy-to-locate place would have been a useful addition

to this book. But that is really the only complaint I have about this well-written and informative book. But what else would you expect from an O'Reilly book?

If nothing else, after reading this book, you should be able to recite by heart the authors' multiple occurrences of driving home the purpose of HTML. Tags in HTML are meant to be used to convey the structure and semantics of a document, not to control the presentation of the document. Clearly, the authors are reacting to the HTML abuse that is rampant on the World Wide Web.

See *<http://www.ora.com>* for more details on this book.

Cultural Treasures of the Internet

by Michael Clark, Prentice-Hall, 1995, ISBN 0-13-209669-2, 313 pp. $22.95

Reviewed by George W. Leach
<gwl1@gte.com>

Internet books are everywhere. Even the bookstores in your local mall have a large collection of titles all aimed at instructing home computer users about the adventures that await them on the Internet. This book is slightly different in its approach. Sure, there are chapters that discuss getting connected online and all of the features of the Internet: email, Usenet, WWW, gopher, etc. And yes, there are lists of resources, which many books do provide. But the resources listed are specifically aimed at the needs of those interested in the arts and humanities. Michael Clark is a professor of English and associate dean of humanities at Widener University in Chester, PA. So his goal in this book is to create a guide for others with the same areas of interest.

The material is organized into four parts: the basics of the Internet, resources, keeping current, and appendices (mailing lists and a glossary).

Part One covers the basics of the Internet. A chapter is devoted to each topic, including "Getting On Line," "Netiquette," "E-mail," "Mailing Lists," "Usenet," "Gopher," "World Wide Web," "Telnet and FTP," and "Archie, Veronica and Jughead." The information is extremely sparse on these topics but does give the reader just enough information to get started. Each chapter ends with a suggested reading section should the reader require more detailed information on the given subject. Did you know there is a book on Netiquette? (Virginia Shea, *Netiquette*, Albion Books, San Francisco, 1994). The book lists for WWW, Gopher, etc. could have been better both in terms of quantity and quality.

Part Two is the meat of the book, covering resources available on the Internet that would be of interest to the target audience. This section accounts for 139 of the 313 pages of this book. The resource directory is organized into categories in alphabetic order. For each resource the author uses an icon to convey the type of access that may be utilized to access the resource: FTP, Telnet, gopher, WWW, etc.

Categories listed include archives and library resources, art and art history, books, education, film, history, humanities, idle pleasures, italian language and literature, fiction online, music, philosophy, poetry, publishers, reference works, electronic texts, theater, travel, US Government, and women's studies. The amount of coverage each category receives varies greatly. For example, the archives and library resources category lists 22 pages of libraries and online library catalogs but poetry doesn't quite take a single page.

The author recognizes the problem with books such as this. The Internet is constantly changing. Any book that attempts to list resources and provide readers with access information is out of date the moment it is completed. Part Three is intended to address this issue by providing information on how to keep up to date on the changing landscape of the information available on the Internet. The author lists over 50 different resources that can be used to keep abreast of changes in just about 20 pages. Among the sources listed are USENET newsgroups and online magazines and newsletters. Unfortunately, there is no mention of any of the major WWW indexing or search sites such as Yahoo, LYCOS, and Webcrawler. And the only major new announcement site listed is the NCSA What's New page.

Part Four consists of two appendices: one devoted to a list of mailing lists of interest to humanities types and a glossary of terms. The mailing lists are organized into categories, although the categories don't always correspond to those used in Part Two of the book. For example, in this section, there is a category called foreign languages with subcategories for specific languages such as Arabic, French, and German. In Part Two of the book there is no foreign languages category. There are categories for French, German, and Italian language and literature. There is another category for languages and literature. The number of categories listed in the appendix is smaller than that in Part Two, but that may be a function of the available mailing lists.

For the target audience of this book, namely, those whose concerns are in the area of the humanities rather than technical in nature, this book provides a decent starting point. In one place, the readers can quickly obtain guidance for accessing information specific to their areas of interest. The coverage of the Internet capabilities and usage provides readers with just enough knowledge to get access to the information listed, which is probably the right level of treatment. Not being involved in the humanities, I don't know how extensive the list of resources could be. Certainly, the information provided to help readers keep current could stand some improvement. But at $22.95, this book is probably worth a look if you are interested in the humanities and want to see what you can access through the Internet.

USENIX 97 Exhibits

At the USENIX Annual Technical and USELINUX Conferences

January 8-9, 1997, Anaheim Marriott Hotel, Anaheim, CA

"A major gathering for the UNIX tribe, (USENIX) focuses on the latest technology and techniques that can be applied immediately." Hot Happening, *ComputerWorld*, 11/20/95

"USENIX meetings are still attended by the breaking edge people in software and systems, but are informal enough that the novices can meet and talk with the more experienced. The meetings have increased in size and have become more diverse, but are still fun, thought provoking, and above all practical." Steve Johnson, *:login;* 6/96

Demonstrate your application development, programming, network management or system administration products and services to the most technically knowledgeable group in computing– USENIX UNIX USERS.

USENIX attendees are sophisticated programmers, developers, system administrators, network managers, engineers, and researchers. When surveyed, they tell us that they are working on, supporting, and developing for many different UNIX and other-than-UNIX platforms. They use UNIX on a daily basis and are committed to the newest tools and technology available. At Anaheim, we are conservatively predicting a gathering of 2000 advanced computing professionals–all of whom are committed to the newest tools and technologies on display in the Exhibit Hall.

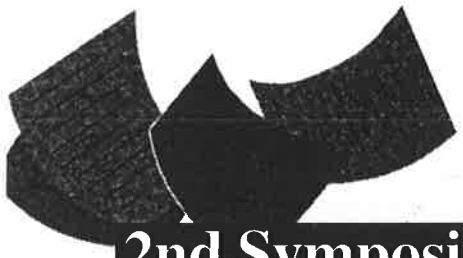
"Two days of exposure to the cream of the UNIX User Community." Neil Groundwater, Enterprise Management Group, SunSoft, Inc.

"My competitors aren't here, and they don't know what they're missing." Brian Duggleby, UNIX Marketing, Digital Equipment Corp.

Companies with reserved space at USENIX '97:

- Addison-Wesley
- Advanced Digital Information Corp.
- AT&T
- Atria Software Inc.
- Auspex Systems, Inc.
- Centon Electronics, Inc.
- Central Design Systems Inc
- Cosmos Engineering Company
- CrossWind Technologies, Inc.
- Digital Equipment Corp.
- Enhanced Software Technologies Inc.
- Enterprise Systems Management Corp.
- Falcon Systems Inc.
- FSA Corporation
- GraphOn Corporation
- McAfee Associates/FSA Corporation
- Max Enhancement Group
- Miller-Freeman, Inc.
- NET-Community
- Network Appliance, Inc.
- O'Reilly & Associates, Inc.
- Parity Systems, Inc.
- PDC
- Prentice Hall PTR
- QMASTER Software Solutions Ltd
- Raima Corporation
- RDI Computer Corp.
- SSC, Inc.
- San Diego Technical Books
- Storage Computer Corporation
- SunSoft, Inc.
- TeamQuest Corporation
- Transitional Technology Inc. (TTI)
- UniTree Software, Inc.
- Western Scientific
- WRQ, Makers of Reflection) Software

Contact Cynthia Deno to reserve exhibit space: 408.335.9445 Email: *<display@usenix.org>*



Announcing USENIX

2nd Symposium on Operating Systems Design and Implementation (OSDI '96)

October 28-31, 1996
Westin Hotel, Seattle, Washington

Sponsored by the USENIX Association.
Co-sponsored by ACM SIGOPS and IEEE TCOS

If you are interested in the newest information on operating systems, plan to attend OSDI. The full program is on our Web site,
<*http://www.usenix.org/osdi96/osdi96.html*>.
You may also send email to <*conference@usenix.org*>, phone 714.588.8649, or fax to 714.588.9706.

Tutorials
Monday, October 28

IPv6: The New Version of the Internet Protocol
Steve Deering, Xerox PARC

Java: A Language for Providing Content on the World Wide Web
Jim Waldo, Sun Microsystems Labs and JavaSoft

Windows NT Internals
Jamie Hanrahan, Kernel Mode Systems

Internet Support for Wireless and Mobile Networking
David B. Johnson, Carnegie Mellon University

Security on the World Wide Web
Daniel E. Geer, Open Market, Inc.

The SimOS Machine Simulation Environment
Mendel Rosenblum, Stanford University

Technical Program
Tuesday, October 29

Invited Talk

JavaOS: Back to the Future
Dr. Jim Mitchell, Sun Fellow, VP of Technology & Architecture, JavaSoft

Technical Sessions

Caching and Prefetching in I/O Systems
Session Chair: Peter Chen, University of Michigan

Automatic Compiler-Inserted I/O Prefetching for Out-of-Core Applications
Todd C. Mowry, Angela K. Demke, and Orran Krieger, University of Toronto

A Trace-Driven Comparison of Algorithms for Parallel Prefetching and Caching
Tracy Kimbrel, University of Washington; Andrew Tomkins and R. Hugo Patterson, Carnegie Mellon University; Brian Bershad, University of Washington; Pei Cao, University of Wisconsin; Edward W. Felten, Princeton University; Garth Gibson, Carnegie Mellon University; Anna R. Karlin, University of Washington; Kai Li, Princeton University

Efficient Cooperative Caching Using Hints
Prasenjit Sarkar and John Hartman, University of Arizona

Issues in Distributed Shared Memory
Session Chair: Marc Shapiro, INRIA

Online Data-Race Detection via Coherency Guarantee
Dejan Perkovic and Pete Keleher, University of Maryland

Lightweight Logging for Lazy Release Consistent Distributed Shared Memory

Manuel Costa, Paulo Guedes, Manuel Sequeira, Nuno Neves, and Miguel Castro, INESC

Performance Evaluation of Two Home-based Lazy Release Consistency Protocols for Shared Virtual Memory Systems
Yuanyuan Zhou, Liviu Iftode, and Kai Li, Princeton University

Works-In-Progress
Session Chair: Karin Petersen, Xerox Parc

Wednesday, October 30

Invited Talk

Active Networks
Dr. David Tennenhouse Associate Professor, Laboratory for Computer Science, Massachusetts Institute of Technology

Scheduling and Synchronization
Session Chair: Kevin Jeffay, University of North Carolina

CPU Inheritance Scheduling
Bryan Ford and Sai R. Susarla, University of Utah

A Hierarchical CPU Scheduler for Multimedia Operating Systems
Pawan Goyal, Xingang Guo, and Harrick M. Vin, University of Texas, Austin

The Synergy Between Non-blocking Synchronization and Operating System Structure
Michael Greenwald and David Cheriton, Stanford University

Operating Systems Abstractions
Session Chair: Richard Draves, Microsoft Research

Microkernels Meet Recursive Virtual Machines
Bryan Ford, Mike Hibler, Jay Lepreau, Patrick Tullman, Godmar Back, Shantanu Goel, and Steven Clawson, University of Utah

Making Paths Explicit in the Scout Operating System
David Mosberger and Larry L. Peterson, University of Arizona

Performance Measurements
Session Chair: Jeff Mogul, Digital Equipment Corporation

Studies of Windows NT Performance Using Dynamic Execution Traces
Sharon E. Perl and Richard L. Sites, DEC SRC

Using Latency to Evaluate Interactive System Performance
Yasuhiro Endo, Zheng Wang, J. Bradley Chen, and Margo Seltzer, Harvard University

Panel Discussion: What the OS Industry Wants from OS Research
Moderator: Jay Lepreau, University of Utah

Thursday, October 31

Extensibility and Safety
Session Chair: Ed Felten, Princeton University

Dynamic Binding for an Extensible System
Przemyslaw Pardyak and Brian Bershad, University of Washington

Dealing with Disaster: Surviving Misbehaved Kernel Extensions
Margo I. Seltzer, Yasuhiro Endo, Christopher Small, and Keith A. Smith, Harvard University

Safe Kernel Extensions Without Run-Time Checking
George C. Necula and Peter Lee, Carnegie Mellon University

Network Interfaces and Protocols
Session Chair: Dave Johnson, Carnegie Mellon University

An Implementation of the Hamlyn Sender-Managed Interface Architecture
Greg Buzzard, David Jacobson, Milon Mackey, Scott Marovich, and John Wilkes, Hewlett-Packard Laboratories

Lazy Receiver Processing (CRP): A New Network Subsystem Architecture for Server Systems
Peter Druschel and Gaurav Banga, Rice University

Effects of Buffering Semantics on I/O Performance
Jose Carlos Brustoloni and Peter Steenkiste, Carnegie Mellon University

Announcing USENIX

2nd USENIX Workshop on Electronic Commerce

November 18-21, 1996
Claremont Resort and Conference Center, Oakland, CA

Tutorial Program
Monday, November 18

Getting Paid on the Internet
Clifford Neuman, University of Southern California

Electronic Payments and Commerce Applications
Taher ElGamal, Netscape Communications Corporation

Secure Java Programming: Fundamentals
Marianne Mueller and David Brownell, JavaSoft

Secure Java Programming: Enhancements
Marianne Mueller and David Brownell, JavaSoft

The Law of Electronic Commerce–Contracts, Records, and Privacy
Benjamin Wright, Attorney and Author

Breaking into the Web (Pun Intended)
Daniel E. Geer, Open Market, Inc.

Technical Program
Tuesday, November 19

Introduction and Welcome
Doug Tygar, Carnegie Mellon University, Program Chair

8:30 am - 10:00 am
Session I: Hardware Tokens
Session Chair: Clifford Neuman, University of Southern California

Tamper Resistance–A Cautionary Note
Ross Anderson, Cambridge University, and Markus Kuhn, Erlanganen/Purdue University

Token-Mediated Certification and Electronic Commerce
Daniel E. Geer, Open Market and Donald T. Davis, SystemExperts

Smart Cards in Hostile Environments
Howard Gobioff, Carnegie Mellon University; Sean Smith, Los Alamos/IBM Research; Doug Tygar, Carnegie Mellon University; Bennet Yee, University of California, San Diego

10:15am - 11:45 am
Session II: Protocol Analysis
Session Chair: Ross Anderson, Cambridge University

Analysis of the SSL 3.0 Protocol
David Wagner, University of California, Berkeley and Bruce Schneier, Counterpane Systems

Fast, Automatic Checking of Security Protocols
Darrell Kindred and Jeannette Wing, Carnegie Mellon University

Verifying Cryptographic Protocols for Electronic Commerce
Randall W. Lichota, Hughes; Grace L. Hammonds, AGCS; Stephen H. Brackin, Arca

1:30 pm - 2:30 pm
Invited Talk: Legal Signatures and Proof in Electronic Commerce
Ben Wright, Attorney and Author

2:45 pm - 4:15 pm
Session III: Policy and Economics
Session Chair: Hal Varian, University of California, Berkeley

Non-Transferable Characteristics and Security Deposits in a Crypto-Economy
Joseph M. Reagle, Consultant

Digital Currency and Public Networks: So What If It Is Secure, Is It Money?
John du Pre Gauntt, London School of Economics

Modeling the Risks and Costs of Digitally Signed Certificates in Electronic Commerce
Ian Simpson, Carnegie Mellon University

4:30 pm - 5:45 pm
Session IV: Standard Payment Interfaces
Session Chair: Bennet Yee, University of California, San Diego

Generic Payment Services: Framework and Functional Specification
Alireza Bahreman, EIT

UPAI: A Universal Payment Application Interface
Steven P. Ketchpel, Hector Garcia-Molina, Andreas Paepcke, Scott Hassan, and Steve Cousins, Stanford University

Payment Method Negotiation Service: Framework and Programming Interface
Alireza Bahreman and Rajkuman Narayanaswamy, EIT

6:00 pm - 8:30 pm
Hosted reception at UC Berkeley

Wednesday, November 20

8:30 am - 10:00 am
Session V: Atomic Transactions
Session Chair: Mark Manasse, DEC Systems Research Center

Anonymous Atomic Transactions
Jean Camp, Sandia National Laboratory, Michael Harkavy, and Doug Tygar, Carnegie Mellon University; Bennet Yee, University of California, San Diego

Strongboxes for Electronic Commerce
Thomas Hardjono and Jennifer Seberry, University of Wollongong

Model Checking Electronic Commerce Protocols
Nevin Heintze, Bell Labs; Doug Tygar, Jeannette Wing, and H. C. Wong, Carnegie Mellon University

10:15 am - 11:45 am
Session VI: Experience
Session Chair: Nathaniel Borenstein, First Virtual Holdings, Inc.

BigDog: Hierarchical Authentication, Session Control, and Authorization for the Web
Benjamin Fried and Andrew Lowry, Morgan Stanley

Financial EDI Over the Internet: Case Study II
Arie Segev, Jaana Porra, and Malu Roldan, University of California, Berkeley

Scalable Document Fingerprinting
Nevin Heintze, Bell Labs

11:45 am - 2:00 pm
Hosted Luncheon with speaker

Designing New Rules of the Road for Electronic Commerce in Digital Information
Pamela Samuelson, University of California, Berkeley

2:00 pm - 3:30 pm
Session VII: Protocols
Session Chair, Daniel E. Geer, Open Market, Inc.

A Protocol for Secure Transactions
Douglas H. Steves, Chris Edmondson-Yurkanan and Mohamed Gouda, University of Texas, Austin

PayTree: "Amortized-Signature" for Flexible MicroPayments
Charanjit Jutla and Moti Yung, IBM

A Peer-to-Peer Software Metering System
Bruce Schneier and John Kelsey, Counterpane Systems

3:45 pm - 6:00 pm
Panel: Electronic Commerce in Practice–What Have We Learned?
Moderator: Clifford Neuman, University of Southern California
Panelists: Nathaniel Borenstein, First Virtual Holdings, Inc.; Marc Briceno, DigiCash; Steve Crocker, Cybercash; Daniel E. Geer, Open Market, Inc.; Arie Segev, University of California, Berkeley; David Van Wie, InterTrust

Thursday, November 21

9:00 am - 10:30 am
Session VIII: Security
Session Chair: Stefan Brands, CWI

Organizing Electronic Services into Security Taxonomies
Sean Smith, Los Alamos National Laboratory/IBM and Paul Pedersen, Los Alamos National Laboratory

WWW Electronic Commerce and Java Trojan Horses
Doug Tygar and Alma Whitten, Carnegie Mellon University

On Shopping Incognito
Ralf Hauser, McKinsey Consulting, Switzerland and Gene Tsudik, University of Southern California, ISI

10:45 am - 11:45 am
Session IX: Software Agents
Session Chair: Doug Tygar, Carnegie Mellon University

Market-Based Negotiation for Digital Library Services
Tracy Mullen and Michael P. Wellman, University of Michigan, Ann Arbor

Information and Interaction in MarketSpace–Towards an Open Agent-Based Market Infrastructure
Joakim Erriksson and Niclas Finne, Telia Research; Sverker Janson, Swedish Institute of Computer Science

Additional Information

To get complete Workshop and registration information, visit the USENIX Web site:
<http://www.usenix.org>
or send email to:
<conference@usenix.org>
or call:
714 588 8549

Announcing USENIX

USENIX 1997Annual Technical Conference

January 6-10, 1997
Anaheim Marriott Hotel, Anaheim, CA

Mark your calendar! Our Annual Technical Conference will provide the latest information and tools to keep you on top of technology. Plus, the first Linux Applications Development and Deployment Conference, USELINUX, will take place at the same time. One fee covers both USENIX and USELINUX conference programs and you can switch freely between them. (Tutorial fees are separate for both.).The full program is available at our Web site, *<http://www.usenix.org>*. You may also send email to *<conference@usenix.org>* or phone 714 588 8649

Early Registration Discount Deadline: November 22
Hotel Discount Deadline: December 20

Tutorial Program
Monday, January 6

Beginning Perl Programming for UNIX Programmers. – **Updated forPerl 5**
Tom Christiansen, Consultant

The Kerberos Approach to Network Security–Updated
Daniel Geer, Open Market, Inc; Jon Rochlis, BBN Planet

An Introduction to Java
Ken Arnold, Sun Microsystems Laboratories

Secure Java Programming–**New**
Marianne Mueller and David Brownell, JavaSoft

Windows NT and Windows 95–The Win32 API–**New**
Joseph M. Newcomer, Consultant

UNIX Network Programming
Richard Stevens, Consultant

Selected Topics in System Administration–**New**
Trent Hein, XOR Network Engineering; Evi Nemeth, University of Colorado, Boulder

How Networks Work–The Limits of Modern Internetworking–**Updated**
Vincent C. Jones, PE

System and Network Performance Tuning–**New**
Hal Stern, Sun Microsystems

Inside the Linux 20 Kernel–**New**
Stephen Tweedie, Digital Equipment Corporation

Tuesday, January 7

UNIX Security Tools: Use and Comparison
Matt Bishop, University of California at Davis

CGI and WWW Programming in Perl–**New**
Tom Christiansen, Consultant

Security on the World Wide Web–**New**
Daniel Geer, OpenMarket, Inc; Jon Rochlis, BBN Planet

Creating Effective User Interfaces–**New**
Joseph A. Konstan, University of Minnesota

Java Applets and the AWT–**New**
Nataraj Nagaratnam, Syracuse University

Setting Up And Administering A Web Server–**New**
Bryan Buus, XOR Network Engineering

Security for Software Developers: How to Write Code that Withstands Hostile Environments–**New**
Marcus J. Ranum, V-ONE Corporation

Solaris System Administration–**New**
Marc Staveley, Consultant

IP version 6: An Introduction
Richard Stevens, Consultant

Writing Device Drivers Under Linux–**New**
Theodore Tso, Massachusetts Institute of Technology

Technical Program
Wednesday, January 8

9:00 am - 10:30 am
Opening Remarks
John Kohl, Pure Atria Corporation

Keynote Address
Developing on "Internet Time"
James Gosling, Sun Microsystems

USELINUX
Linux: What It Is and Why It Is Significant
Mark Bolzern, Work Group Solutions; Tom Miller, X Engineering Software Systems

11:00 am - 12:30 pm
Performance I
Session Chair: Carl Staelin, Hewlett-Packard Laboratories

Embedded Inodes and Explicit Grouping: Exploiting Disk Bandwidth for Small Files
Gregory R. Ganger and M. Frans Kaashoek, Massachusetts Institute of Technology

Observing the Effects of Multi-Zone Disks
Rodney Van Meter, University of Southern California, Information Sciences Institute

A Revisitation of Kernel Synchronization Schemes
Christopher Small and Stephen Manley, Harvard University

Invited Talk
Nomadicity and the IETF
Charles E. Perkins, IBM T.J. Watson Research Center

USELINUX
The Sparc Port of Linux
David S. Miller, Rutgers CAIP; Miguel de Icaza,Instituto de Ciencias Nucleares, Ciudad Universitaria, Universidad Nacional Autonoma de Mexico

2:00 pm - 3:30 pm
Interface Tricks
Session Chair: Rob Gingell, Sun Microsystems

Porting UNIX to Windows NT
David G. Korn, AT&T Research

Protected Shared Libraries–A New Approach to Modularity and Sharing
Arindam Banerji, John M. Tracey, and David L. Cohn, University of Notre Dame

A Novel Way of Extending the Operating System at the User-Level: The Ufo Global File System
Albert D. Alexandrov, Maximilian Ibel, Klaus E. Schauser, and Chris J. Scheiman, University of California, Santa Barbara

Invited Talk
If Cryptography Is So Great, Why Isn't It Used More?
Matt Blaze, AT&T Research

USELINUX
Advanced Device Drivers
Alessandro Rubini, Universitá di Pavia

4:00 pm - 5:00 pm
Client Tricks
Session Chair: Fred Douglis, AT&T Research

Network-Aware Mobile Programs
Mudumbai Ranganathan, Anurag Acharya, Shamik Sharma, and Joel Saltz, University of Maryland

Using Smart Clients to Build Scalable Services
Chad Yoshikawa, Brent Chun, Paul Eastham, Amin Vahdat, Thomas Anderson, and David Culler, University of California, Berkeley

Invited Talk
The Inktomi Web Search Engine
Eric Brewer, University of California, Berkeley

USELINUX
4:00 pm - 5:30 pm
Future of the Linux Kernel
Linus Torvalds, Helsinki University

Thursday, January 9

9:00 am - 10:30 am
Clustering
Session Chair: Clem Cole, Digital Equipment Corporation

Building Distributed Process Management on an Object-Oriented Framework
Ken Shirriff, Sun Microsystems Laboratories

Adaptive and Reliable Parallel Computing on Networks of Workstations
Robert D. Blumofe, University of Texas, Austin, and Philip A. Lisiecki, Massachusetts Institute of Technology

A Distributed Shared Memory Facility for FreeBSD
Pedro A. Souto and Eugene W. Stark, State University of New York, Stony Brook

Invited Talk
The AltaVista Web Search Engine
Louis Monier, Digital Equipment Corporation

USELINUX
Real Time
Victor Yodaiken and Michael Barabanov, New Mexico Institute of Technology

11:00 am - 12:30 pm
Tools
Session Chair: Matt Blaze, AT&T Research

Libcdt: A General and Efficient Container Data Type Library
Kiem-Phong Vo, AT&T Research

A Simple and Extensible Graphical Debugger
David R. Hanson and Jeffrey L. Korn, Princeton University

Cget, Cput, and Stage–Safe File Transport Tools for the Internet
Bill Cheswick, Bell Laboratories

Invited Talk
IPv6: The New Version of the Internet Protocol
Steve Deering, Xerox Palo Alto Research Center

USELINUX
/proc
Stephen Tweedie, Digital Equipment Corporation

The Pluggable Authentication Modules (PAM) Framework
Ted T'so, Massachusetts Institute of Technology

2:00 pm - 3:30 pm
Works in Progress
Session Chair: John Schimmel, Silicon Graphics, Inc.

Invited Talk
Highlights from 1996 USENIX Conferences and Workshops

USELINUX
Standards
Heiko Eissfeldt, Unifix Software

4:00 pm - 5:30 pm
Joint Session Inferno
Rob Pike, Bell Labs

USELINUX
Connecting Legacy and Open Systems
Michael Callahan, Stelias Computing, Inc.

Friday, January 10

9:00 am - 10:30 am
User Something
Session Chair: Nathaniel Borenstein, First Virtual Holdings

WebGlimpse–Combining Browsing and Searching
Udi Manber, Michael Smith, and Burra Gopal, University of Arizona

Mailing List Archive Tools
Sam Leffler and Melange Tortuba, Silicon Graphics, Inc.

Experience with GroupLens: Making Usenet Useful Again
Bradley N. Miller, John T. Riedl, and Joseph A. Konstan, University of Minnesota

Invited Talk
Measuring Computer Systems: How to Tell the Truth with Numbers
Margo Seltzer and Aaron Brown, Harvard University

USELINUX
Linux: What It Is and Why It Is Significant
Mark Bolzern, Work Group Solutions; Tom Miller, X Engineering Software Systems

Linux and Distribution Channels: Ways to Enter the Commercial Market
Don Rosenberg, Stromian Technologies

11:00 am - 12:30 pm
Performance II
Session Chair: Mike Karels, Berkeley Software Design

Overcoming Workstation Scheduling Problems in a Real-Time Audio Tool
Isidor Kouvelas and Vicky Hardman, University College London

On Designing Lightweight Threads for Substrate Software
Matthew Haines, University of Wyoming

High-Performance Local-Area Communication With Fast Sockets
Steven H. Rodrigues, Thomas E. Anderson, and David E. Culler, University of California, Berkeley

Invited Talk
Stupid Net Tricks
Bill Cheswick, Bell Laboratories

USELINUX Business
Using Linux in Your Business: A Business Justification
Presented by Linux International

2:00 pm - 3:30 pm
Caching and Stashing
Session Chair: Bill Bolosky, Microsoft Research

An Analytical Approach to File Prefetching
Hui Lei and Dan Duchamp, Columbia University

Optimistic Deltas for WWW Latency Reduction
Gaurav Banga, Fred Douglis, and Michael Rabinovich, AT&T Research

A Toolkit Approach to Partially Connected Operation
Dan Duchamp, Columbia University

Invited Talk
Finding Bugs in Concurrent Programs
Gerard J. Holzmann, Bell Laboratories

USELINUX Business
2:00 pm - 4:00 pm
The Linux Market: Who, What, Where, When, and Why?
Member of the Board, Linux International

4:15 pm - 5:45 pm
Joint Closing Session
Severe Tire Damage's Stupid Mbone Tricks–A Lecture/Demonstration

The Seventh Conference on Computers, Freedom, and Privacy: Commerce & Community

March 11-14, 1997
San Francisco Airport Hyatt Regency Hotel, Burlingame, CA

Call for Participation

CFP97: Commerce & Community will be sponsored by the Association for Computing Machinery, SIGCOM, and SIGSAC. The host institutions will be Stanford University and the University of California, Berkeley. Co-sponsors and cooperating organizations include the ACM SIGCAS, the Electronic Frontier Foundation, the Center for Democracy and Technology, the Electronic Privacy Information Center, and the WELL.

CFP97: Commerce & Community is the latest in a series of annual conferences assembling a diverse group of experts and advocates from the domains of technology, business, government, and academia to explore the evolution of information and communication technologies and public policy, and its effects on freedom and privacy in the United States and throughout the world.

Past CFP sessions have discussed, debated–and often anticipated–issues of great social import. In this tradition, CFP97: Commerce & Community will examine the social and policy questions posed by:

- the growth of electronic communities
- electronic commerce and the commercialization of cyberspace
- the problems of legal and regulatory control of the Net
- the interests of privacy and property in the electronic domain
- high-tech law enforcement and security concerns.

The CFP97 Program Committee invites your suggestions for presentations on these or other important issues at the nexus of technology, business, public policy, freedom, and privacy.

Proposals may be for individual talks, panel discussions, debates, moot courts, moderated, interactive sessions, or other formats. Each proposal should be accompanied by a one-page statement describing the topic and format. Descriptions of multi-person presentations should include a list of proposed participants and session chair. Proposals should be sent by email to <*cfp97@cfp.org*>. If necessary, typewritten proposals may be sent to: CFP97, 2210 Sixth Street, Berkeley, CA 94710.

Please submit your proposal as soon as possible. The deadline for submissions is October 1, 1996. (Please note that we have extended our deadline for submissions.)

For more information on the Computers, Freedom and Privacy Conferences, as well as up-to-date announcements on CFP97, please visit our Web page at:< *http://www.cfp.org*>.

Announcement and Call for Papers USENIX

3rd Conference on Object-Oriented Technologies and Systems (COOTS '97)

June 16-19, 1997
Portland Mariott Hotel, Portland, OR

Sponsored by the USENIX Association

Important Dates

Tutorial submissions due: *February 6, 1997*
Paper submissions due: *February 12, 1997*
Notification to authors: *February 25, 1997*
Camera-ready final papers due: *May 6th, 1997*

Preliminary Program Committee

Program Chair: Steve Vinoski, *Hewlett-Packard*
Tutorial Program Chair: Douglas C. Schmidt, *Washington University*
Don Box, *DevelopMentor*
David Chappell, *Chappell & Associates*
David Cohn, *University of Notre Dame*
Jim Coplien, *Lucent Bell Labs*
Murthy Devarakonda, *IBM TJ Watson Research Center*
Daniel Edelson, *IA Corporation*
Rachid Guerraoui, *EPFL*
Doug Lea, *SUNY Oswego*
Dmitry Lenkov, *Hewlett-Packard*
Mark Linton, *Vitria*
Stan Lippman, *Walt Disney Feature Animation*
Igor Metz, *GLUE Software Engineering*
Rajendra Raj, *Morgan Stanley*
Ron Resnick, *Nortel*
Vince Russo, *Purdue Univirsity*
Jonathan Shopiro, *Novell*
Joe Sventek, *Hewlett-Packard Laboratories*
Ozan Yigit, *Border Network Technologies*
Jim Waldo, *JavaSoft*

Overview

COOTS '97 is intended to showcase advanced R&D work in object-oriented technologies and software systems. The conference emphasizes experimental research and experience gained by using object-oriented techniques and languages to build complex software systems that meet real world needs.

Tutorials

COOTS '97 will begin with two days of tutorials. We expect tutorial topics to include:

- Distributed object systems (CORBA DCOM, etc.)
- Java and WWW programming languages
- Object-oriented network programming
- Design patterns for object-oriented systems
- Concurrent object-oriented programming
- Efficient and effective framework design
- Evolution of ANSI/ISO C++ standardization
- C++ Standard Template Library
- Alternative object-oriented languages

Tutorial proposal submissions must be received by *February 6, 1997*. Send via email to: Doug Schmidt *<schmidt@cs.wustl.edu>*. Tutorials selected for presentation at the conference will be announced by February 19, 1997

Conference Topics

Two days of technical sessions will follow the tutorials. We seek papers describing original work concerning the design, implementation, experimentation, and use of object-oriented technologies. COOTS emphasizes advanced engineering aspects of object technology, focusing on experimental systems research and development on distributed objects, multimedia, operating systems, compiler technology, Java, and C++. While papers covering work in Java and C++ are strongly encouraged, the conference is broader in scope than its predecessors. In particular, we invite submissions describing results and work in other object-oriented or object-based languages. Potential topics include but are not limited to:

- Applications of, and experiences with, object-oriented technologies in various domains (distributed systems, multimedia, real-time systems, financial services, human/computer interface, etc.)
- Distributed object systems (CORBA, DCOM, the Web.)
- Implementations of commercial object infrastructures and reliable distributed objects (ORB Plus, Orbix, NextStep, DCOM, DSOM, Isis/RDO, Distributed Smalltalk, Java ORBs, etc.)
- Object-oriented programming language development environments and tools (C++, Java, Smalltalk, Modula-3, Eiffel, Ada95, etc.)
- Content-oriented languages for programmin in the WWW (Java, Python, Obliq, Phantom, etc.)
- Interface description languages (OMG IDL, Microsoft IDL, etc.)
- C++ standardization (STL, templates, implementation challenges)

Questions regarding a topic's relevance to the workshop may be addressed to the program chair via electronic mail to *<vinoski@apollo.hp.com>*. Proceedings of the workshop will be published by USENIX and will be provided free to technical ses-

sion attendees; additional copies will be available for purchase from USENIX.

Awards will be given to the best paper and to the best student paper at the conference.

Advanced Topics Workshop

This year's conference will conclude with an Advanced Topics Workshop, an informal setting in which to exchange in-depth technical information with your peers. It is open to authors of accepted papers and to participants who submit position papers related to the workshop's topic. The topic will be determined several months prior to the conference and a Call for Position papers will be announced.

What To Submit

Technical paper submissions must be received by *February 12th, 1997*. Full papers should be 10 to 15 pages (5,000-6,000 words). In lieu of full papers, authors may submite extended abstracts which should be 5-7 pages (2,500-3,500 words), not counting references and figures. The body of the extended abstract should be written in complete paragraphs, and convince reviewers that a good, solid paper and presentation will result. Extended abstracts are intended to stimulate industrial participation and to allow publication of very current material.

All submissions will be judged on originality, relevance, and correctness. Each accepted submission will be assigned a member of the program committee to shepherd it through to the final paper. Camera-ready final papers are due *May 6, 1997*.

Please include a cover letter stating the paper title and authors along with the name of the person who will act as the contact to the program committee. Please include a surface mail address, daytime and evening phone number, and, if available, an email address and fax number for the contact person. If all of the authors are students, please indicate that in the cover letter for consideration of the $500 "Best Student Paper" award.

The COOTS conference, like most conferences and journals, requires that papers not be submitted simultaneously elsewhere and may not have been previously published or be subsequently published. Papers accompanied by "Non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Detailed guidelines for submission and examples of extended abstracts are available from the USENIX office at 510 528 8649, or email *<cootsauthors@usenix.org>*, or *<vinoski@apollo.hp.com>*. An electronic version of the Call for Papers is available at *<http://www.usenix.org>*.

Where To Submit

Please send one copy of a full paper or an extended abstract to the program committee via one of the following methods. All submissions will be acknowledged.

Preferred Method: email (Postscript or ASCII) to:
<cootspapers@usenix.org>

Alternate Method: postal delivery to:

Steve Vinoski (program chair)
USENIX COOTS Conference
Hewlett-Packard, MS CHR-03-WR
300 Apollo Drive
Chelmsford, MA 01824
Tel: 508 436 5904
•Fax: 508 436 5122

Registration Information

Details of the technical and tutorial programs, registration fees and forms and hotel information will be available beginning in April 1997. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630
Phone: 714 588. 8649
Fax: 714. 588. 9706
Email: *conference@usenix.org*
URL: http://www.usenix.org

Or you can send email to our mailserver at *<info@usenix.org>*. Your message should contain the line: send catalog. A catalog will be sent to you.

Announcement and Call for Participation USENIX

11th Systems Administration Conference (LISA 97)

October 26 - 31, 1997
Town & Country Hotel, San Diego, CA

Co-sponsored by USENIX, the Advanced Computing Systems Professional and Technical Association, and SAGE, the System Administrators Guild

Refereed paper submissions dates

Extended abstracts due:	June 3, 1997
Notification to authors by:	June 30, 1997
Final papers due:	September 9, 1997
Registration materials available:	July, 1997

Program Co-Chairs

Hal Pomeranz, NetMarket/CUC International
Celeste Stokely, Stokely Consulting

Program Committee

Paul Anderson, University of Edinburgh
Melissa Binde, Swarthmore College
Helen E. Harrison, SAS Institute, Inc.
Trent R. Hein, XOR Network Engineering
Amy Kreiling, SAS Institute, Inc.
William LeFebvre, Group sys Consulting
Dinah McNutt, Sysadmazon, Inc.
Adam S. Moskowitz, Interval Research Corp.
Wendy Nather, Swiss Bank Warburg
John Sellens, University of Waterloo

Invited Talks Coordinators

Rik Farrow, Internet Security Consulting
Pat Wilson, Dartmouth College

LISA, the Systems Administration Conference, is the leading conference for and by system administrators. LISA originally stood for "Large Installation Systems Administration" when a large installation meant over 100 users, 100 systems, or a gigabyte of disk storage. Today, LISA offers the most comprehensive program for system administrators from sites of all sizes and at all levels of experience.

LISA has always been the best source for the technology, tools, and techniques needed to keep pace with today's rapid advancements. As we move into our second decade, LISA also recognizes that a system administrator needs professional skills which extend beyond the purely technical. The theme for LISA 97 is Professional Development, and the conference will include additional information and programming designed to enhance all aspects of your career.

TUTORIAL PROGRAM

Sunday - Tuesday, October 26-28, 1997

Whether you are a novice or senior system administrator, you will be able to find a tutorial at your level. There will be up to 35 tutorials over three days.

To provide the best possible tutorial offerings, USENIX continually solicits proposals for new tutorials. If you are interested in presenting a tutorial at this or other USENIX conferences, please contact the tutorial coordinator:

Daniel V. Klein
phone: 412 421 0285
Fax: 412 421 2332
Email: <dvk@usenix.org>

TECHNICAL SESSIONS

Wednesday - Friday, October 29-31, 1997

You will have two different types of learning opportunities during the technical sessions: a track dedicated to refereed technical and professional development papers and one for invited talks, panels and Works-in-Progress (WIP) reports.

CONFERENCE TOPICS

Papers and presentations addressing the following topics are particularly timely. Presentations addressing other areas of general interest are equally welcome.

Technology, Tools, and Techniques

- Innovative system administration tools and techniques
- Distributed or automated system administration
- Integration of emerging technologies
- Incorporation of commercial system administration technology
- Experiences supporting large sites (>1000 users or machines)
- Experiences supporting nomadic and wireless computing
- Intranet development, support, and maintenance
- Integrating new networking technologies
- Integration of heterogeneous platforms including legacy systems
- Managing enterprise-wide email
- OS / platform migration strategies
- Performance analysis and monitoring
- Data management
- Security

The Business of System Administration

- Customer service skills
- Problem tracking
- Proactive problem management
- Asset management

- Support strategies in use at your site
- Delivering effective presentations
- Effective training techniques for system administration and users
- Managing others and managing your boss
- Working effectively with contractors or consultants
- Working as a contractor or consultant
- Ethics

REFEREED PAPER SUBMISSIONS

An extended abstract of two to five pages is required for the paper selection process. Full papers are not acceptable at this stage; if you send a full paper, you must also include an extended abstract.

Include references to establish that you are familiar with related work, and, where possible, provide detailed performance data to establish that you have a working implementation or measurement tool.

Submissions will be judged on the quality of the written submission, and whether or not the work advances the state of the art of system administration. For more detailed author instructions and a sample extended abstract, send email to
<lisa97authors@usenix.org>
or call USENIX at 510 528 8649.

Note that LISA, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication, and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned unread. All submissions are held in the highest confidence prior to publication in the conference proceedings, both as a matter of policy and as protected by the US Copyright Act of 1976.

Authors of an accepted paper must provide a final paper for publication in the conference proceedings. At least one author of each accepted paper presents the paper at the conference. Final papers are limited to 20 pages, including diagrams, figures and appendices, and must be in troff, ASCII, or LaTeX format. We will supply you with instructions. Papers should include a brief description of the site, where appropriate.

Conference proceedings, containing all refereed papers and materials from the invited talks, will be distributed to attendees and will also be available from USENIX following the conference.

WHERE TO SEND SUBMISSIONS

Please submit extended abstracts for the refereed paper track by two of the following methods:

- Email to:
  <lisa97papers@usenix.org>
- Fax to: 510 548 5738
- Mail to:
  LISA 97 Conference
  USENIX Association
  2560 Ninth Street, Suite 215
  Berkeley CA USA 94710

To discuss potential submissions, and for inquiries regarding the content of the conference program, contact the program co-chairs at *<lisa97chair@usenix.org>* or contact:

- Hal Pomeranz, NetMarket/CUC International
  155 Second Street
  Cambridge MA 02141
  Phone: 617 441 5050, x316
  Fax: 617 441 5099
  Email: *<hal@usenix.org>*

- Celeste Stokely, Stokely Consulting
  211 Thompson Square
  Mountain View CA 94043
  Phone: 415 967 6898
  Fax: 415 967 0160
  Email: *<celeste@usenix.org>*

CASH PRIZES

Cash prizes will be awarded at the conference for the best paper and the best student paper.

INVITED TALK TRACK

If you have a topic of interest to system administrators, but not suited for a traditional technical paper submission, please submit a proposal for an invited talk to the invited talk (IT) coordinators at *<itlisa@usenix.org>*.

VENDOR DISPLAYS
Wednesday and Thursday October 29 - 30, 1997.
LISA attendees have an enormous interest in industrial strength, state-of-the-art solutions to system administration problems. If your company's products provide solutions, LISA will provide attendees with the technical expertise to understand and appreciate them. Please contact:

- Cynthia Deno
  Phone: 408 335 9445
  Fax: 408 335 5327
  Email: *<display@usenix.org>*

BIRDS-OF-A-FEATHER SESSIONS

Birds-of-a-Feather sessions (BoFs) are very informal gatherings of attendees interested in a particular topic.. BoFs are held Tuesday, Wednesday, and Thursday evenings. BoFs may be scheduled in advance by phoning the Conference Office. They may also be scheduled at the conference.

REGISTRATION INFORMATION

Complete program and registration information will be available in July 1997. To receive registration materials, please contact:

- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA 92630
  Phone: 714 588 8649
  Fax: 714 588 9706
  Email: *<conference@usenix.org>*
  URL: *<http://www.usenix.org>*

Or email the line: send catalog to our mailserver at *<info@usenix.org>*. A catalog will be returned to you.

Announcement and Call for Participation USENIX

7th USENIX Security Symposium

January 26-29, 1998
Marriott Hotel–San Antonio, Texas

Sponsored by the USENIX Association in cooperation with: the CERT Coordination Center.

Important Dates for Refereed Papers

Papers due:	September 9, 1997
Author notification:	October 8, 1997
Camera-ready final papers due:	December 9, 1997
Registration materials available:	End October, 1997

(Authors, see "How to Submit a Refereed Paper," below.)

Program Chair

Avi Rubin, Bellcore

Program Committee

Carlisle Adams, Nortel
Dave Balenson, Trusted Information Systems
Steve Bellovin, AT&T Research
Dan Boneh, Bellcore
Diane Coe, Concepts Technologies
Ed Felten, Princeton University
Li Gong, JavaSoft
Peter Honeyman, CITI, University of Michigan
Hugo Krawczyk, IBM Watson Labs
Jack Lacy, AT&T Research
Hilarie Orman, DARPA/ITO
Mike Reiter, AT&T Research
David Wagner, University of California, Berkeley

Readers

Katherine T. Fithen, CERT
Trent Jaeger, IBM Watson Labs

Invited Talks Coordinator

Greg Rose, Qualcomm

Conference Home Page

<http://www.usenix.org/sec/sec98.html>

Overview

The goal of this symposium is to bring together researchers, practitioners, system programmers, and others interested in the latest advances in security and applications of cryptography.

This will be a four day symposium with two days of tutorials, followed by two days of refereed paper presentations, invited talks, work in progress presentations, and panel discussions.

Tutorials:
Monday and Tuesday, January 26-27

Tutorials for both technical staff and managers will provide immediately useful, practical information on topics such as local and network security precautions, what cryptography can and cannot do, security mechanisms and policies, firewalls and monitoring systems.

If you are interested in proposing a tutorial, contact the tutorial coordinator, Dan Klein: phone 412 421 2332 ; email *<dvk@usenix.org>*.

Technical Sessions
Wednesday and Thursday, January 28-29

In addition to the keynote presentation, the technical program includes refereed papers, invited talks, a work in progress session, and panel sessions. There will be Birds-of-a-Feather sessions the last two evenings. You are invited to make suggestions to the program committee via email to *<security@usenix.org>*.

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the symposium proceedings, published by USENIX and provided free to technical session attendees. Additional copies will be available for purchase from USENIX.

Symposium Topics

Refereed paper submissions are being solicited in areas including but not limited to:

- Adaptive security and system management
- Analysis of malicious code
- Applications of cryptographic techniques
- Attacks against networks/machines
- Computer misuse and anomaly detection
- Copyright protection (technical solutions)
- Cryptographic & other security tools
- File and file system security
- Network security
- New firewall technologies
- Security in heterogeneous environments
- Security incident investigation and response
- Security of Mobile Code
- User/system authentication
- World Wide Web security

Note that this symposium is not about new codes, ciphers, nor cryptanalysis for its own sake.

Papers must represent novel scientific contributions in computer security with direct relevance to the engineering of secure systems for the commercial sector.

How to Submit a Refereed Paper

Please read carefully

The guidelines for submission are a bit different from previous years. Authors must submit a mature paper in PostScript format. Any incomplete sections (there shouldn't be many) should be outlined in enough detail to make it clear that they could be finished easily. Full papers are encouraged, and should be about 8 to 15 typeset pages. Submissions must be received by September 9, 1997.

Along with your paper, please submit a separate email message containing the title, all authors, and their complete contact information (phone, fax, postal address, email), including an indication of which author is the contact author.

Authors will be notified of acceptance on October 8, 1997.

All submissions will be judged on originality, relevance, and correctness. Each accepted submission may be assigned a member of the program committee to act as its shepherd through the preparation of the final paper. The assigned member will act as a conduit for feedback from the committee to the authors. Camera-ready final papers are due on December 9, 1997.

If you would like to receive detailed guidelines for submission and examples of extended abstracts, you may send email to: *<security-authors@usenix.org>* or telephone the USENIX Association office at 510 528 8649.

The Security Symposium, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Where To Submit

For reliability, please send one copy of your paper to the program committee via each of two of the following methods. All submissions will be acknowledged.

Preferred method: email (PostScript) to:

<securitypapers@usenix.org>

Alternate method: postal delivery to:

Security Symposium
USENIX
2560 Ninth St., Suite #215
Berkeley CA 94710
USA

Phone: 510 528 8649
Fax: 510 548 5738

Vendor Exhibits

Demonstrate your security product to our technically astute attendees responsible for security at their sites. We invite you to take part in the Vendor Display. The informal, table-top display allows you to meet with attendees informally and demonstrate in detail your security solutions, contact:

Cynthia Deno
Email: *<cynthia@usenix.org>*
Phone: 408 335 9445
Fax: 408 335 5327

Works in Progress Session (WIP)

The last session of the symposium will be a Works-in-Progress session consisting of 5 minute presentations. Speakers should provide a one or two paragraph abstract to the program chair by 6:00 pm on January 28, 1998 at the conference. These should be provided in person, not via email. The chair will post the schedule of presentations by noon on the 29th. Experience at other conferences has shown that usually, all of them are accepted. The 5 minute time limit will be strictly enforced.

Invited Talks

There will be several invited talks at the conference in parallel with the refereed papers. If you have suggestions for possible speakers, please send them to *<security@usenix.org>*.

Registration Materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available at the end of October 1997. To receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA USA 92630
Phone: 714 588 8649
Fax: 714 588 9706
Email: *<conference@usenix.org>*

Information can also be found under the USENIX URL: *<http://www.usenix.org>*.

15th IEEE SYMPOSIUM ON RELIABLE DISTRIBUTED SYSTEMS

October 23-25, 1996
Niagara-on-the-Lake, Canada

THEME: The focus of the symposium concerns the building and use of dependable distributed and parallel systems, particularly with system properties such as reliability, availability, and performance. In addition to traditional papers, the published papers deal with experimental results, testbeds, development, and data from operational systems. We focus on timely papers that describe industrial projects, experimental systems, exploratory applications in large networks, and topics related to the next generation of large reliable distributed systems.

For more information about the conference and registration, visit the SRDS-15 web site at:
<http://www.crhc.uiuc.edu/srds96/>

This year's SRDS is being held in conjunction with the HASE High-Assurance Systems Engineering Workshop on October 21-22, 1996. Visit the HASE web site at *<http://rodin.cs.uh.edu/~hase96/>*

- **SYMPOSIUM CHAIR:** David Taylor, U. of Waterloo
- **PROGRAM COMMITTEE CHAIR:** W. Kent Fuchs, U. of Illinois
- **LOCAL ARRANGEMENTS CHAIR:** Jay Black, U. of Waterloo
- **FINANCE CHAIR:** Hanan Lutfiyya, U. of Western Ontario
- **PUBLICITY CHAIR:** Ron Bianchini, Carnegie Mellon U.
- **REGISTRATION CHAIR:** Debbie Mustin, U. of Waterloo
- **TC LIAISON:** Bharat Bhargava, Purdue U.

PROGRAM COMMITTEE:

- A. Abouelnaga (TRW)
- F. Bastani (U. of Houston)
- R. Chillarege (IBM)
- M. Dal Cin (U. of Erlangen)
- J. Bechta Dugan (U. of Virginia)
- E. N. Elnozahy (Carnegie Mellon U.)
- P. Ezhilchelvan (U. of Newcastle)
- A. Goscinski (Deakin U.)
- P. Guedes (INESC, Lisbon)
- R. Harper (Stratus)
- A. Helal (Purdue U.)
- F. Jahanian (U. of Michigan)
- P. Jalote (IIT, Kanpur)
- T. Kikuno (Osaka U.)
- K. Kim (U. of California, Irvine)
- C. Kintala (AT&T Bell Labs)
- M. Lyu (AT&T Bell Labs)
- Y. Min (Academia Sinica, P. R. China)
- K. Mori (Hitachi)
- E. Nett (GMD)
- D. Powell (LAAS)
- C. Pu (Oregon Graduate Institute)
- W. Sanders (U. of Illinois)
- L. Simoncini (U. of Pisa)
- A. Tai (SoHaR)
- J. Tang (Memorial U., Newfoundland)

SPONSORS: IEEE Computer Society Technical Committees on Distributed Processing, Fault-Tolerant Computing, and Operating Systems

Advance Technical Program Schedule
Wednesday, October 23, 1996

9:00am Opening Session

10:30am Session 1: Novel Architectures

Chair: K. Kim

Exploiting Data-Flow for Fault-Tolerance in a Wide-Area Parallel System
A. Nguyen-Tuong, A. S. Grimshaw, and M. Hyett (University of Virginia, Department of Computer Science)

Specialized N-Modular Redundant Processors in Large-Scale Distributed Systems
I-L. Yen (Michigan State University, Department of Computer Science)

Dynamic Fault Tolerance in DCMA - A Dynamically Configurable Multicomputer Architecture
H. Kuefner and H. Baehring (FernUniversitaet Hagen, Department of Computer Science)

1:30pm Session 2: Evaluation & Testing

Chair: K. Mori

Analysis of a Multistage Interconnection Network using Binary Decision Diagrams (BDD)
J. Chiu (National Semiconductor Corporation) and J. Bechta Dugan (University of Virginia, Department of Electrical Engineering)

Analyzing Dynamic Voting Using Petri Nets
I.-R. Chen and D.-C. Wang (Institute of Information Engineering, National Cheng Kung University, Taiwan)

On-line Testing for Application Software of Widely-Distributed Systems
E. Nishijima, H. Yamamoto (both of the Systems Development Laboratory, Hitachi, Ltd.), and K. Fujiwara (Omika Works, Hitachi, Ltd.)

Thursday, October 24, 1996

8:30am Session 3: Checkpointing and Stable Storage

Chair: E. N. Elnozahy

Minimizing Timestamp Size for Completely Asynchronous Optimistic Recovery with Minimal Rollback
S. W. Smith (Computer Research and Applications Group, Los Alamos National Laboratory) and D. B. Johnson (Carnegie Mellon University, School of Computer Science)

Improving the Performance of Coordinated Checkpointers on Networks of Workstations using RAID Techniques
J. S. Plank (University of Tennessee, Department of Computer Science)

Implementation and Performance of a Stable Storage Service in Unix
F. Cristian (University of California, San Diego, Department of Computer Science and Engineering), S. Mishra, and Y. S. Hyun (both of University of Wyoming, Department of Computer Science)

10:30am Session 4: Diagnosis

Chair: M. Dal Cin

Hierarchical Adaptive Distributed System-Level Diagnosis Applied for SNMP-based Network Fault Management
E. P. Duarte Jr. and T. Nanya (Tokyo Institute of Technology, Graduate School of Information Sciences)

Dynamic Fault Diagnosis
by W. Hurwood (Yale University)

Diagnosing Crosstalk-Faulty Switches in Photonic Switching Networks
C. Qiao (State University of New York at Buffalo, Department of Electrical and Computer Engineering)

1:30pm Session 5: Group Communication

Chair: C. Kintala

A Transparent Light-Weight Group Service
L. Rodrigues (Instituto de Engenharia de Sistemas e Computadores (INESC)), K. Guo (Cornell University, Computer Science Department), A. Sargento (INESC), R. van Renesse (Cornell University, Computer Science Department), B. Glade (Isis Distributed Systems Division, Stratus Computer Inc.), P. Veri'ssimo (INESC), and K. Birman (Cornell University, Computer Science Department)

Strong and Weak Virtual Synchrony in Horus
R. Friedman and R. van Renesse (Cornell University, Department of Computer Science)

The Design of a CORBA Group Communication Service
P. A. Felber, B. Garbinato, and R. Guerraoui (Ecole Polytechnique Fédérale de Lausanne, Département d'Informatique)

3:30pm Panel: Industry Perspective on Reliable Distributed Systems

R. Harper - Organizer and Moderator Panelists: Y. Huang (System and Software Research Center, Bell Labs, Lucent Technologies) R. Chillarege (IBM TJ Watson Research Center) P. Green (Stratus Computer, Inc.) S. Low (Tandem Computers)

Friday, October 25, 1996

8:30am Session 6: Replication Issues

Chair: M. Lyu

Locating More Corruptions in a Replicated File
A. W.-c. Fu (The Chinese University of Hong Kong, Department of Computer Science) and S.-C. Chau (University of Lethbridge (Alberta, Canada), Department of Mathematics and Computer Science)

Primary Copy Method and its Modifications for Database Replication in Distributed Mobile Computing Environment
A. Zaslavsky, M. Faiz, B. Srinivasan, A. Rasheed, and S. J. Lai (M. Faiz from SPL Worldgroup (S) Pte Ltd, Singapore. All others from Monash University (Melbourne), Department of Computer Technology)

A Fault-Tolerant CORBA Name Server
S. Maffeis (Cornell University and Olsen & Associates, Zurich)

10:30am Session 7: Time Constraints

Chair: J. Tang

Fail-Aware Failure Detectors
C. Fetzer and F. Cristian (University of California, San Diego, Department of Computer Science & Engineering)

A Causal Message Ordering Scheme for Distributed Embedded Real-Time Systems
K. M. Zuberi and K. G. Shin (University of Michigan, Department of Electrical Engineering and Computer Science)

A Proposal for Ensuring High Availability of Distributed Multimedia Applications
M. Billot, V. Issarny, I. Puaut, and M. Banâtre (IRISA, France)

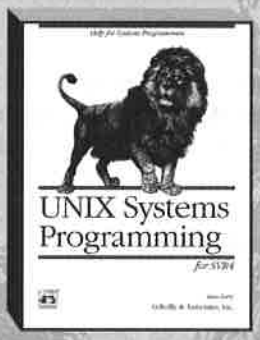
You can register on-line at:
<http://www.crhc.uiuc.edu/srds96/>

10% Discount to USENIX Members on all O'Reilly Book Titles.
Sorry, discount does not apply to O'Reilly software products.

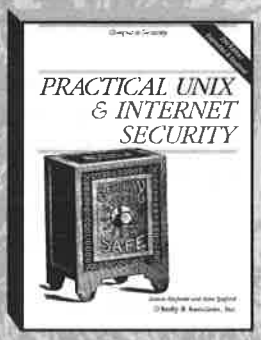
MASTERY

LIKE EXPERTS PEERING OVER YOUR SHOULDER,
OUR BOOKS GIVE YOU REAL-WORLD ADVICE

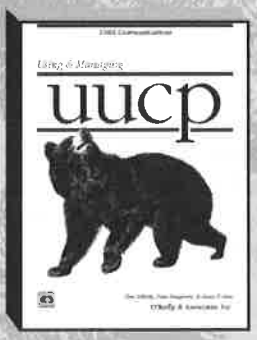
Mastering a topic means going beyond the mechanics and gaining a thorough understanding of the problem you're trying to solve. It's this kind of thinking that's behind all our books; you'll gain a "big picture" perspective and will learn, at the same time, time-saving, practical tips and tricks the experts use.



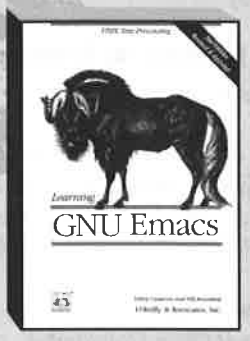
UNIX Systems Programming for SVR4
By David A. Curry
1st Edition Summer 1996
640 pages (est.), ISBN 1-56592-163-1
$34.95 (est.)



Practical UNIX & Internet Security, 2nd Edition
By Simson Garfinkel & Gene Spafford
2nd Edition April 1996
1004 pages, ISBN 1-56592-148-8
$39.95



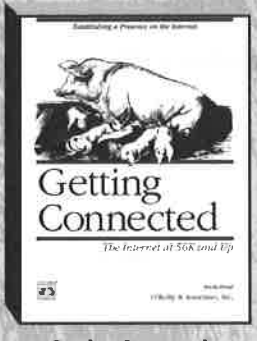
Using & Managing UUCP
By Ed Ravin, Tim O'Reilly, Dale Dougherty & Grace Todino
1st Edition Summer 1996
350 pages (est.), ISBN 1-56592-153-4
$29.95 (est.)



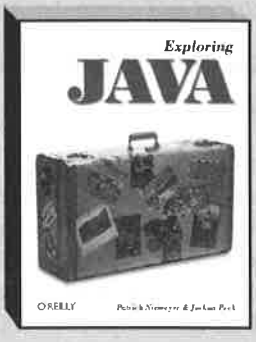
Learning GNU Emacs, 2nd Edition
By Debra Cameron & Bill Rosenblatt
2nd Edition Summer 1996
450 pages (est.), ISBN 1-56592-152-6
$29.95 (est.)

Running Linux, 2nd Edition
By Matt Welsh & Lar Kaufman
2nd Edition Summer 1996
650 pages (est.), ISBN 1-56592-151-8
$24.95

Running Linux Companion CD-ROM
By O'Reilly & Assoc. & Red Hat Software
2nd Edition Summer 1996
120 pages (est.), ISBN 1-56592-212-3
$24.95 (est.), Includes two CD-ROMs



Getting Connected
By Kevin Dowd
1st Edition June 1996
424 pages, ISBN 1-56592-154-2
$29.95



Exploring Java
By Pat Niemeyer & Josh Peck
1st Edition May 1996
426 pages, ISBN 1-56592-184-4
$29.95

101 Morris Street, Sebastopol, CA 95472 • FAX: 707-829-0104
Credit card orders: 800-889-8969 Weekdays 6AM-5PM PST
Please mention code ALOG when ordering
For inquiries: 800-998-9938, 707-829-0515

O'REILLY™
PUBLISHING · SOFTWARE · RESEARCH

To request a copy of our catalog: **catalog@online.ora.com**
For complete descriptions of all our titles, check out:
http://www.ora.com/
Also available at local bookstores.

Computer Publications from John Wiley & Sons, Inc.

USENIX members receive a 15% discount

The Internet Navigator, 2ed
Paul Gilster
1-05260-4 $24.95 **member price: $21.20**
# of copies: _____

Advanced Topics in UNIX
Ronald Leach
1-03663-3 $24.95 **member price: $21.20**
# of copies: _____

Introduction to Client Server Systems
Paul Renaud
1-57774-X $34.95 **member price: $29.70**
# of copies: _____

Portable UNIX
Douglas Topham
1-57926-2 $14.95 **member price: $12.71**
# of copies: _____

UNIX, Self-Teaching Guide
George Leach
1-57924-6 $19.95 **member price: $16.95**
# of copies: _____

Object Oriented Programming with Turbo C++
Keith Weiskamp
1-52466-2 $24.95 **member price: $21.20**
# of copies: _____

Obfuscated C and Other Mysteries
Don Libes
1-57805-3 $39.95 **member price: $33.96**
# of copies: _____

Finding It On the Internet
Paul Gilster
1-03857-1 $19.95 **member price $16.95**
# of copies: _____

Internationalization: Developing Software for Global Markets 1-07661-9 ***(pub. date: 1/95)***
Tuoc Luong $29.95 **member price: $25.45**
# of copies: _____

Adventures in UNIX Network Applications Programming
Bill Rieken
1-52858-7 $39.95 **member price: $33.96**
# of copies: _____

UNIX Shell Programming, 3e
Lowell Jay Arthur
1-59941-7 $29.95 **member price: $25.45**
# of copies: _____

The UNIX Command Reference Guide
Kaare Christian
1-85580-4 $32.95 **member price: $28.01**
# of copies: _____

Berkeley UNIX: A Simple & Comprehensive Guide
James Wilson
1-61582-X $40.95 **member price: $34.80**
# of copies: _____

☐ Payment enclosed, plus sales tax
☐ VISA ☐ Mastercard
☐ American Express
Card No. _____
Name _____
Firm _____
Address _____
City/State/Zip _____
Signature _____
(order invalid unless signed

Please send all orders to:

John Wiley & Sons, Inc.
Attn: Karen Cooper, Special Sales
605 Third Avenue
New York, NY 10158
Phone: (212) 850-6789
Fax: (212) 850-6142

NETWORKS

USENIX members receive a 15% discount on the following MIT Press publications:

GLOBAL NETWORKS
Computers and International Communication

edited by Linda M. Harasim

Global Networks takes up the host of issues raised by the new networking technology that now links individuals, groups, and organizations in different countries and on different continents. The twenty-one contributions focus on the implementation, application, and impact of computer-mediated communication in a global context.

340 pp. $29.95 hardcover HARNH

THE NETWORK NATION
Human Communication via Computer
Revised Edition

Starr Roxanne Hiltz and Murray Turoff

"*The Network Nation*... contained a fascinating vision. ...It is a place where thoughts are exchanged easily and democratically and intellect affords one more personal power than a pleasing appearance does. Minorities and women compete on equal terms with white males, and the elderly and handicapped are released from the confines of their infirmities to skim the electronic terrain as swiftly as anyone else." — Teresa Carpenter, *Village Voice*

580 pp. $24.95 paperback HILWP

THE EVOLUTION OF C++
Language Design in the Marketplace of Ideas

edited by Jim Waldo

This collection of articles traces the history of C++, from its infancy in the Santa Fe workshop, to its proliferation today as the most popular object-oriented language for microcomputers. Waldo notes in his postscript that in the process of evolving, the language has lost a clearly articulated, generally accepted design center, with no common agreement about what it should or should not do in the future.

279 pp. $24.95 paperback WALEP

TECHNOLOGY 2001
The Future of Computing and Communications

edited by Derek Leebaert

Researchers, executives, and strategic planners from inside the companies and laboratories that have shaped today's information age forecast the merging technologies that could well define the computing and communications environment that lies ahead.

392 pp. $14.95 paperback LEEEP

THE DIGITAL WORD
Text-Based Computing in the Humanities

edited by George P. Landow and Paul Delany

This book explores the larger realm of the knowledge infrastructure where texts are received, reconstructed, and sent over global networks.

Technical Communication and Information Systems series 384 pp. $39.95 hardcover LANDH

SOCIOMEDIA
Multimedia, Hypermedia, and the Social Construction of Knowledge

edited by Edward Barrett

Sociomedia continues the assessment of hypertext and hypermedia systems begun in *Text, ConText, and HyperText* and *The Society of Text*. It examines the use of integrated multimedia to support social or collaborative research, learning, and instruction in the university, one of the best environments for developing and analyzing the effects of computing technologies on our understanding of complex sets of information.

Technical Communications and Information Series 360 pp. $50.00 hardcover BARRH

CONNECTIONS
New Ways of Working in the Networked Organization

Lee Sproull and Sara Kiesler

"...Sproull and Kiesler raise crucial questions about our technical and particularly our human strategies as a producing society."
— Howard Webber, *Sloan Management Review*

228 pp. $21.95 paperback SPRCP

TECHNOBABBLE

John A. Barry

"A serious study of the language of the new technocracy."
— William Safire, *The New York Times Magazine*

288 pp. $12.50 paperback BARCP

Please send me these titles _____

☐ Payment enclosed ☐ Purchase Order Attached Charge to my: ☐ Master Card ☐ VISA

Card # _____ exp. _____ Signature _____

$ _____ Total for book(s)
$ 3.00 Postage for North American addresses
$ _____ Canadian customers add 7% GST
$ _____ Total for book(s) & postage

Name _____
Address _____
City _____ State _____ Zip _____
Phone _____ FAX _____

Make checks payable and send order to:

THE MIT PRESS
55 Hayward Street, Cambridge, MA 02142-1399 USA

To order by phone, call (617) 625-8569 or (800) 356-0343. E-mail order # mitpress-orders@mit.edu. The operator will need this code: **UNIX1.**

Prentice Hall PTR is pleased to recommend the following titles to USENIX members...

USENIX members receive a 15% discount on orders

____**UNIX System Administration Handbook, Second Edition,** Evi Nemeth/Garth Snyder, 0-13-151051-7
(15105-0) *List:* $48.00 *Members:* $40.80

____**Object-Oriented Modeling and Design,** James Rumbaugh, 0-13-629841-9
(62984-0) *List:* $54.00 *Members:* $45.90

____**Zen and the Art of the Internet, Third Edition,** Brendan Kehoe, 0-13-121492-6
(12149-1) *List:* $23.95 *Members:* $20.36

____**The Magic Garden Explained,** Bernard Goodheart/James Cox, 0-13-098138-9
(09813-7) *List:* $38.00 *Members:* $32.30

____**Internetworking with TCP/IP, Vol. II Design, Implementation, and Internals,** Douglas E. Comer/David L. Stevens, 0-13-472242-6
(47224-1) *List:* $61.33 *Members:* $52.13

____**SCO Performance Tuning Handbook,** Gina Miscovich/David Simons, 0-13-102690-9
(10269-9) *List:* $42.00 *Members:* $35.70

____**Object-Oriented Programming,** Peter Coad/Jill Nicola, 0-13-032616-X
(03261-5) *List:* $48.00 *Members:* $40.80

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the BSD Socket Version,** Douglas E. Comer and David L. Stevens, 0-13-474222-2
(47422-1) *List:* $53.00 *Members:* $45.05

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the AT&T TLI Version,** Douglas E. Comer and David L. Stevens, 0-13-474230-3
(47423-9) *List:* $53.00 *Members:* $45.05

____**The Internet Message: Closing the Book with Electronic Mail,** Marshall T. Rose, 0-13-092941-7
(09294-0) *List:* $50.00 *Members:* $42.50

____**The Standard C Library,** PJ Plauger, 0-13-131509-9
(13150-8) *List:* $37.80 *Members:* $32.13

____**All About Administering the NIS+, Second Edition** Rick Ramsey, 0-13-309576-2
(30957-5) *List:* $42.00 *Members:* $35.70

____**The Simple Book: An Introduction to Internet Management** Marshall T. Rose, 0-13-177254-6
(17725-3) *List:* $55.00 *Members:* $46.75

____**Networking Operations on UNIX SVR4,** Mike Padavano, 0-13-613555-2
(61355-4) *List:* $50.00 *Members:* $42.50

____**Solaris Porting Guide,** Sunsoft ISV Engineering 0-13-030396-8
(03039-5) *List:* $52.00 *Members:* $44.20

____**Multiprocessor System Architectures,** Ben Catanzaro 0-13-089137-1
(08913-6) *List:* $44.00 *Members:* $37.40

____**The HP-UX System Administrator's "How To" Book** Marty Poniatowski, 0-13-099821-4
(09982-0) *List:* $32.00 *Members:* $27.20

____**UNIX System V Performance Management,** Edited by Phyllis Eve Bregman and Sally A. Browning 0-13-016429-1
(01642-8) *List:* $29.95 *Members:* $25.45

____**SCO® UNIX® Operating System System Administrator's Guide,** Santa Cruz Operation, 0-13-012568-7
(01256-7) *List:* $39.00 *Members:* $33.15

HERE'S HOW TO ORDER:

CALL
800-880-6818

OR WRITE:
CompuBooks
Route 1, Box 271-D,
Cedar Creek, TX 78612

OR INTERNET:
70007.1333@CompuServe.com
(GO CBK on CompuServe)

WE SHIP ANYWHERE!

FOR MORE INFORMATION, OR QUANTITY ORDERS, PLEASE CALL 201-592-2657

A UNIQUE OFFER ON THE BEST IN UNIX FOR USENIX MEMBERS

20% DISCOUNT FROM McGRAW-HILL

☐ **THE INTERNET GUIDE FOR NEW USERS**
D. Dern
hardcover, 016510-6, $40.00,
MEMBER PRICE $32.00
paperback, 016511-4, $27.95,
MEMBER PRICE $22.36

☐ **INTERNET FOR EVERYONE**
R. Wiggins
hardcover, 067018-8, $29.95,
MEMBER PRICE $23.96
paperback, 067019-6, $45.00,
MEMBER PRICE $36.00

☐ **THE ESSENTIAL INTERNET INFORMATION GUIDE**
J. Manger
707905-1, paperback, $27.95,
MEMBER PRICE $22.36

☐ **THE INFORMATION BROKERS HANDBOOK, Second Edition**
S. Rugge
911878-x, paperback, $34.95,
MEMBER PRICE $27.96
Available December 1994

☐ **SAA AND UNIX: IBM's Open System Strategy**
M. Killen
034607-0, $40.00,
MEMBER PRICE $32.00

☐ **A STUDENT'S GUIDE TO UNIX**
H. Hahn
025511-3, paperback, $28.00,
MEMBER PRICE $22.40

☐ **UNIX DEVELOPER'S TOOL KIT**
K. Leininger
911836-4, $65.00,
MEMBER PRICE $52.00

☐ **UNIX SECURITY: A Practical Tutorial**
N. Arnold
002560-6, $24.95,
MEMBER PRICE $19.96

☐ **THE UNIX AUDIT: Using UNIX to Audit UNIX**
M. Grottola
025127-4, $32.95,
MEMBER PRICE $26.36

☐ **UNIX: A Database Approach**
S. Das
015745-6, $29.95,
MEMBER PRICE $23.96
Available November 1994

I am a member of USENIX Association. Please send me the books I have indicated at the member special rate. I have added $3.00 postage and handling for the first book ordered, $1.00 for each additional book, plus my local sales tax.

Check or money order is enclosed—payable to McGraw-Hill, Inc.
Charge my ❑ Visa ❑ Mastercard ❑ Discover ❑ Amex

Account #_____

Expiration Date_____

Signature_____

USENIX Membership # _____

Bill & Ship To:

Name_____

Street_____

City, State, Zip_____

Daytime Phone #_____

03US002

Send or Fax Orders to:

McGraw-Hill, Inc.
Attn: Rosa Perez
11 West 19th Street—4th Floor
New York, New York 10011
Fax: 212-337-4092

If I am not completely satisfied, I will return the book(s) within 15 days for a full refund or credit. Satisfaction unconditionally guaranteed. Prices subject to change without notics. We can only accept orders within the continental USA.

8th Annual

OPEN SYSTEMS WORLD®
FedUNIX '96

Including

- ✔ **Network Security '96 Conference**
  The UNIX Network Security Conference
- ✔ **Motif '96**
  The 6th Annual Motif/X/CDE International Users Conference
- ✔ **The 3rd Annual Linux Users' & Developers' Conference**
- ✔ **1st Annual IntraNetworking Summit**
- ✔ **Creating the Electronic Government** *FREE Sessions including JAVA, UNIX-NT Interoperability, Mobile Computing, and more!*
- ✔ **5th Annual Federal Technology Leadership Awards,** *Sponsored by Government Executive Magazine*
- ✔ **Great Courses & Instructors**
- ✔ **Trade Show**
  with over 125 exhibitors including: HP, IBM, SCO, DG, Platinum Technology, Landmark, BMC, Digital, Hummingbird, Cabletron, WRQ and many others

OPEN SYSTEMS WORLD
◆ FedUNIX '96 ◆

Conference:
Nov. 4–8, 1996

Exhibition:
Nov. 6–7, 1996

Washington Convention Center
Washington, D.C.

Please Send Me:
❑ Conference Information ❑ Exhibitor Information
❑ Tutorial Information ❑ Federal Leadership Awards
❑ Exposition Information

Name _____
Organization _____
Address _____
_____
Telephone _____
Fax: _____ E mail _____

Return to:
OPEN SYSTEMS WORLD
10440 Shaker Drive, Suite 203
Columbia, MD 21046
FAX: 301-596-8803
Email: oswinfo@wauug.org

FOR INFORMATION CALL **301-596-8800**

UN

Conferences/Tutorials
Mon. Nov. 4–Fri. Nov. 8

Exhibition
Wed. Nov. 6–Thurs. Nov. 7

Washington Convention Center
Washington, D.C.

Call
301-596-8800
for more information

Email: oswinfo@wauug.org

LOCAL USER GROUPS

The Association will support local user groups by doing a mailing to assist in the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to: *<login@usenix.org>*.

California

Fresno: The Central California UNIX Users Group has a WWW contact page to which members may post questions or information. For connection information:

- Steve Mitchell
  209 278 5675
  <http://warpig.cati.csufresno.edu/ccuug/ccuug.html>

Orange County: Meets the 2nd Monday of each month

- UNIX Users Association of Southern California (UUASC)
  Dave Close
  714 434 7359
  <dave@uuasc.org>

Colorado

Boulder: Meets monthly at different sites; for membership information and meeting schedule, send email to <fruug-info@fruug.org>.

- Front Range UNIX Users Group
  Lone Eagle Systems Inc.
  636 Arapahoe #10
  Boulder, CO 80302
  Steve Gaede
  303 444 9114
  <gaede@fruug.org>
  <http://www.fruug.org/~fruug>

Washington, D.C.

Meets 2nd Thursday of each month.

- Washington Area UNIX Users Group
  10440 Shaker Drive, Suite 103
  Columbia, MD 21046
  301 621 5500
  email:*<afedder@wauug.org>*

Florida

Orlando: Meets the 3rd Thursday of each month.

- Central Florida UNIX Users Group
  Bob Boarman
  <bboardman@national.aaa.com>

Western: Meets 1st Thursday of each month.

- Florida West Coast UNIX Users Group
  Mike Delucia
  813 882 0770
  <pfl@cftnet.com>

Georgia

Atlanta: Meets on the 1st Monday of each month in White Hall, Emory University.

- Atlanta UNIX Users Group
  P.O. Box 12241
  Atlanta, GA 30355-2241
  Mark Landry 404 365 8108

Kansas and Missouri

Meets on 2nd Tuesday of each month.

- Kansas City UNIX Users Group (KCUUG)
  P.O. Box 412622
  Kansas City, MO 64141
  816 891 1093
  <richj@northcs.cps.com>

Michigan

Detroit/Ann Arbor: Meets on the 2nd Thursday of each month in Ann Arbor.

- Southeastern Michigan Sun Local Users Group and Nameless UNIX Users Group
  Steve Simmons
  office: 313 769 4086
  home: 313 426 8981
  <scs@lokkur.dexter.mi.us>

Missouri

St. Louis:

- St. Louis UNIX Users Group
  P.O. Box 2182 St. Louis, MO 63158
  Terry Linhardt
  314 772 4762
  <uunet!jgaltstl!terry>

New England

Northern New England UNIX Users Group (NNEUUG)
Meets monthly at different sites.

- Peter Schmitt 508 289 2877
  Woods Hole Oceanographic Institute
  Woods Hole, MA
  <pschmitt@whoi.edu>

New Mexico

Albuquerque: ASIGUNIX <asigunix@rt66.com>meets every 3rd Wednesday of each month.

- Phil Hortz 505 275 0466
  <prh@bossnet.com>

New York

New York City: Meets every other month in Manhattan.

- Unigroup of New York City
  G.P.O. Box 1931
  New York, NY 10116
  <uniboard@unigroup.org>
  J. P. Radley 212 877 0440

Oklahoma

Tulsa: Meets 2nd Wednesday of each month.

- Tulsa UNIX Users Group,
  $USR Bill Hunt 918 494 4848
  <bhunt@tulsix.utulsa.edu>
  Mark Lawrence 918 749 7498
  <lawrence@tulsix.utulsa.edu>

Texas

Austin: Meets 3rd Thursday of each month.

- Capital Area Central Texas UNIX Society (CACTUS)
  P.O. Box 9786
  Austin, TX 78766-9786
  Ronald S. Woan
  512 838 1254
  <president@cactus.org>
  <http://cactus.org>

Dallas/Fort Worth: Meets the 1st Thursday of each month.

- Dallas/Fort Worth UNIX Users Group
  P.O. Box 867405
  Plano, TX 75086
  Evan Brown 214 519 3577

Houston: Meets 3rd Tuesday of each month.

- Houston UNIX Users Group (Hounix)
  answering machine:
  713 684 6590
  Jack Gilbert, President
  713 862 3637
  <jack@hounix.org>

Washington

Seattle: Meets monthly.

- Seattle UNIX Group
  Bill Campbell 206 947 5591
  P.O. Box 820
  Mercer Island, WA 98040-0820
  <slug@seaslug.org>
  <http://www.seaslug.org>

Canada

Calgary: Meets 4th Tuesday of each month.

- Calgary UNIX Users Group (CUUG)
  David Marwood
  <postmaster@cuug.ab.ca>
  <http://www.cuug.ab.ca:8001>

Manitoba: Meets 2nd Tuesday of each month.

- Manitoba UNIX User Group (MUUG)
  P.O. Box 130
  St. Boniface Winnipeg,
  MB R2H 3B4
  Bary Finch, President
  204 934 1690
  <info@muug.mb.ca>

Ottawa: Meets 3rd Wednesday of each month except July and August

- The Ottawa Carleton UNIX Users Group (OCUUG)
  Dave Blackwood
  <dave@revcan.ca>

Toronto:

- 143 Baronwood Court
  Brampton, Ontario
  Canada L6V 3H8
  Evan Leibovitch
  416 452 0504
  <evan@telly.on.ca>

Quebec: Meets first Wednesday every 3rd month.

- Administrateurs de Système
  UNIX du Quebec (ASUQ)
  Université de Montreal,
  Dept. IRO
  C.P. 6128, Succ. Centre-Ville
  Montreal, Quebec, Canada,
  H3C 3J7
  514 343 7480

System Administration Groups

Back Bay LISA (BBLISA)

New England forum covering all aspects of system and network administration, for large and small installations. Meets monthly, at MIT in Cambridge, MA.

For information, contact:
- J. R. Oldroyd 617 227 5635
  <jr@opal.com>
- Mailing list subscription:
  <bblisa-request@bblisa.org>
- Mailing list postings:
  <bblisa@bblisa.org>
- For current calendar of events:
  finger *<bblisa@finger.bblisa.org>*

Bay LISA (California)

Meets 3rd Thursday of each month at Cisco in Mountain View, CA. For more information, please contact:
<info@baylisa.org>
or visit:
<http://www.baylisa.org>
or FTP:
<ftp.baylisa.org:/BayLISA/location>

dc.sage (Metropolitan Washington, D.C.)

"Users can be a friend of the system administrator, but they will never be able to be a peer." We're here to meet, interact, support, leverage, and otherwise make your vocation a more fruitful one. For more information, send "info dc-sage" to:

<majordomo@mrj.com>.
or contact::

Carolyn J. Sienkiewicz
<cjs@chokey.mo.md.us>

Brad Knowles
<bknowles@aol.net>

$GROUPNAME (New Jersey)

$GROUPNAME is an organization in New Jersey formed to facilitate information exchange pertaining to the field of UNIX system administration. For more information, send "infogroupname" to
<majordomo@plts.org>.
or visit
<http://www.groupname.org>

Tom Limoncelli *<tal@big.att.com>*

New York System Administrators (NYSA)

NYSA, the place where New York area system administrators can do that other kind of "networking," meets the second Tuesday of every month. Receive meeting announcements by emailing the word "subscribe" in the body of the message to *<majordomo@nysa.org>*. For more information visit our web site *<http://www.nysa.org>*

Phone: 201 692 1638

North Carolina System Administrators Group

The North Carolina System Administrators Group meets on the 2nd Monday each month around the Research Triangle Park area.

- Amy Kreiling 919 962 1843
  <kreiling@cs.unc.edu>
- William E. Howell 919 941 4868
  <william_howell@glaxo.com>

Seattle SAGE Group (SSG)

A group for system administrators in the Seattle and Northwestern Washington areas. We are somewhat UNIX-centric, but many members have experience with other networked OSs. We meet the 2nd Thursday of each month. Contact Scottie Swenson: *<swenson@u.washington.edu>*.

Twin Cities System Administrators (TCSA)

TCSA meets on the 3rd Thursday of each month in the Twin Cities area of Minnesota.

<http://www.tcsa.org>
<info@tcsa.org>

ACM: Association for Computing Machinery

ALE: Application Language Engineering Conference

ASPLOS: Architectural Support for Programming Languages and Operating Systems

CFP: Conference on Computers, Freedom, and Privacy

COOTS: Conference on Object-Oriented Technologies and Systems

DECUS: Digital Equipment Computer Users Society

HotOS: Hot Topics in Operating Systems

ICDCS: Int'l Conference on Distributed Computing Systems

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force

ISOC: Internet Society

IWOOOS: International Workshop on Object-orientation in Operating Systems

JUS: Japan UNIX Society

LISA: USENIX/SAGE Systems Administration Conference

OOPSLA: Object-oriented Programming Systems, Languages and Applications

OSDI: Symposium on Operating Systems Design & Implementation

POPL: Principles of Programming Languages

ROSE: Open Systems in Romania

SANS: System Administration, Networking & Security

SIGCOMM: Data Communication

SIGPLAN: ACM Special Interest Group on Programming Languages

SOSP: ACM Symposium on Operating Systems Principles

UniForum: International Association of UNIX and Open Systems Professionals

CALENDAR OF EVENTS

This is a combined calendar of conferences, symposia, and standards meetings. If you ha an event that you wish to publicize, please contact <*login@usenix.org*>. For complete USENIX conference and symposia listings see URL <*http://www.usenix.org/events/general.html*>.
For an up-to-date, comprehensive, and easy-to-access information resource on the Internet, covering events all over the world, consult the WWW Virtual Library on Conferences at Fraunhofer-IAO.
<*http://www.rpd.net/Info/conferences*>

1996

October

1 - 4 ASPLOS VII, Cambridge, MA
7 - 11 NetWorld+Interop '96, Paris, France
8 - 10 UNIX Expo, New York City
10 - 16 OOPSLA '96, San Jose, CA
23 - 25 IEEE Symposium on Reliable Distributed Systems, Niagara, Canada
27 - 28 *IWOOOS '96, Seattle, WA
28 - 31 *OSDI II, Seattle, WA
28 - Nov. 2 ROSE '96, Bucharest, Romania

November

4 - 8 Open Systems World/ FedUNIX Washington, DC
4 - 8 UNIX Network Security, Washington, DC
9 - 14 DECUS, Anaheim, CA
17 - 22 ACM IEEE-CS Supercomputing '96, Pittsburgh, PA
18 - 20* Electronic Commerce, Oakland, CA

December

9 - 13 IETF, San Jose, CA

1997

January

6 - 10 *USENIX, Anaheim, CA
6 - 10 *USELINUX Conference, Anaheim,CA
20 - 24 POPL '97

February

10 - 11 ISOC Symposium on Network & Distributed Systems

March

1 - 5 ACM '97, San Jose, CA
10 - 14 UniForum, San Francisco, CA
11 - 14 CFP '97, Burlingame, CA

April

7 - 11 IETF, Memphis, TN
21 - 26 SANS, Baltimore, MD

May

5 - 7 HotOS-VI
27 - 30 ICDCS '97, Baltimore, MD

June

16 - 19 *COOTS III, Portland, OR
16 - 20 SIGPLAN '97

July

*5th Annual Tcl/Tk Workshop

August

3 - 8 SIGGRAPH '97, Los Angeles, CA

September

14 - 18 SIGCOMM '97, Cannes, France
16 - 18 UNIX & Windows NT Expo, NY

October

5 - 8 SOSP, St. Malo, France
5 - 9 OOPSLA '97
15 - 17 *ALE, Santa Barbara, CA
26 - 31 *LISA '97, San Diego, CA

1998

January

19 - 23 POPL '98
26 - 29 *7th USENIX Security Symposium

June

15 - 19 *USENIX, New Orleans, LA

October

18 - 22 OOPSLA '98

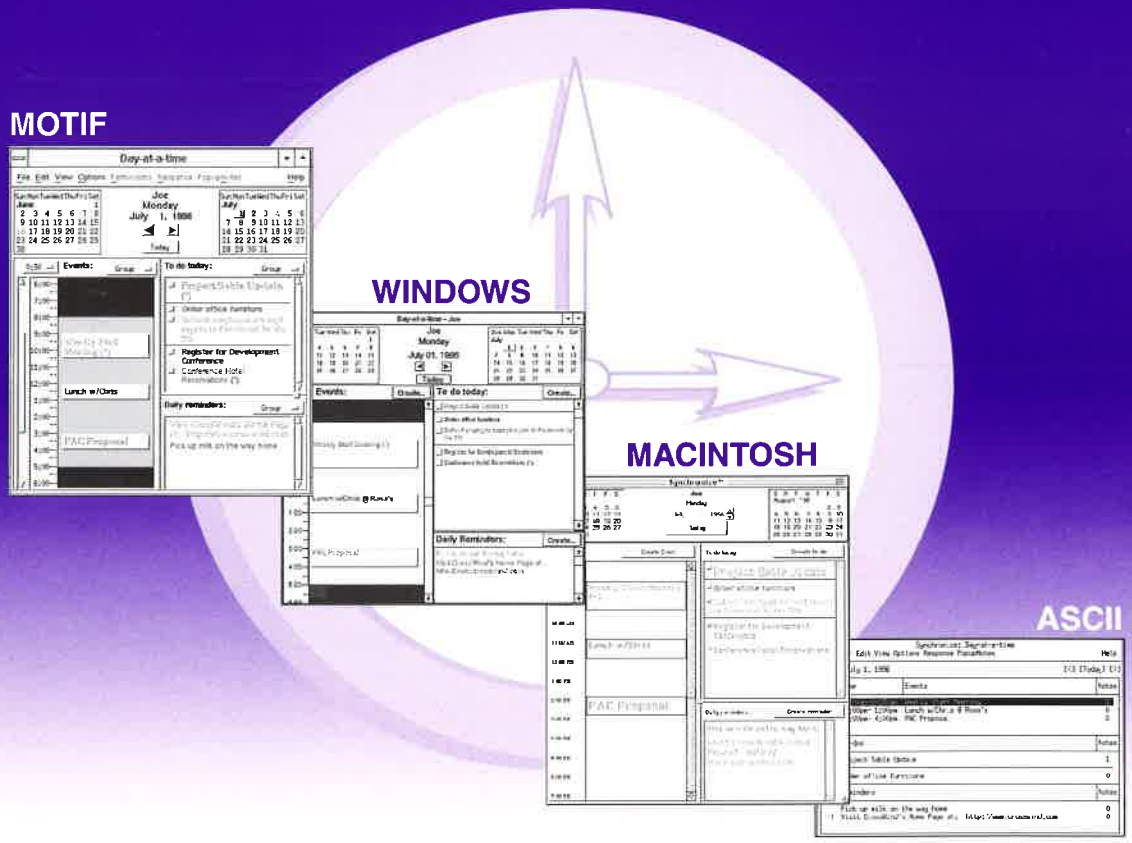
December

7 - 11 *LISA '98, Boston, MA

* = events sponsored by the USENIX Association.

FINALLY...

Cross-Platform Scheduling/Calendaring that's as easy to administer as it is easy to use!



SYNCHRONIZE™

Synchronize gives you the scheduling, task and resource management features your users want -- without the installation and administration headaches you may have encountered with other workgroup software.

Enterprise-wide scalabilty to support thousands of users
Today, employees work in groups that cross and transcend departmental boundaries and geographic locations. Synchronize's client/server architecture with distributed databases technology enables diverse workgroups in multiple time zones to work together as a tight, close-knit team...with peak efficiency and effective communication. And Synchronize scales transparently allowing you to easily add new users as your enterprise grows.

Real-time data access
Because Synchronize communicates directly across TCP/IP to transfer data, information access is virtually instantaneous. The latency associated with file-based or email dependent access methods and slower transport protocols is eliminated.

Cross-platform support
No scheduling software supports more platforms than Synchronize. Designed for cross-platform deployments, Synchronize runs on over 20 commercial UNIX and NT server platforms and includes desktop clients for Windows, Macintosh, X.11/Motif and ASCII terminals.

Take Synchronize for a test drive--FREE!
Can Synchronize solve your scheduling and task mana problems...increase workgroup productivity...and communication across your organization? To help you we'd like to send you a FREE 30-Day Evaluation.

Try Synchronize Risk-Free
No Obligation--Nothing to Return
Call Today!
(800) 335-4988
info@crosswind.com

See Synchronize at:
LISA '96 - Booth #23
October 2-3, Chicago
UNIX EXPO - Booth #830
October 8-10, New York

CROSSWIND TECHNOLOGIES
Software for the Cooperative Workplace
http://www.crosswind.com

PERIODICALS POSTAGE
PAID
AT BERKELEY, CALIFORNIA
AND ADDITIONAL OFFICES

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

POSTMASTER:
SEND ADDRESS CHANGES
TO *;login:*
USENIX ASSOCIATION
2560 NINTH STREET
SUITE 215
BERKELEY, CA 94710